ADAPTIVE CHANNEL EQUALIZATION USING RADIAL BASIS FUNCTION NETWORKS AND MLP

SHEEJA K.L.

Department of Electrical Engineering

National Institute of Technology Rourkela

Rourkela-769008, Orissa, India.

ADAPTIVE CHANNEL EQUALIZATION USING RADIAL BASIS FUNCTION NETWORKS AND MLP

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Technology

In

ELECTRONICS SYSTEMS AND TELECOMMUNICATION

By

SHEEJA K.L.

Department of Electrical Engineering

National Institute Of Technology

Rourkela

2009

ADAPTIVE CHANNEL EQUALIZATION USING RADIAL BASIS FUNCTION NETWORKS AND MLP

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Technology

In

ELECTRONICS SYSTEMS AND TELECOMMUNICATION

By

SHEEJA K.L.

Under the Guidance of

Prof. SANJEEB MOHANTY

Department of Electrical Engineering

National Institute Of Technology

Rourkela

2009

National Institute Of Technology
Rourkela

CERTIFICATE

This is to certify that the thesis entitled, **"Adaptive Channel Equalization Using Radial Basis Function Networks and MLP"** submitted by Mrs.**Sheeja K.L.** in partial fulfillment of the requirements for the award of Master of Technology Degree in **Electrical Engineering** with specialization in **"Electronics Systems and Telecommunication"** at the National Institute of Technology, Rourkela (Deemed University) is an authentic work carried out by her under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University / Institute for the award of any Degree or Diploma.

Prof. Sanjeeb Mohanty
Dept. of Electrical Engineering
National Institute of Technology
Rourkela-769008

Date:

ACKNOWLEDGEMENTS

This project is by far the most significant accomplishment in my life and it would be impossible without people who supported me and believed in me.

I would like to extend my gratitude and my sincere thanks to my honorable, esteemed supervisor **Prof. Sanjeeb Mohanty,** Department of Electrical Engineering. He is not only a great lecturer with deep vision but also most importantly a kind person. I sincerely thank him for his exemplary guidance and encouragement. His trust and support inspired me in making right decisions and I am glad to have worked with him.

I want to thank all my teachers **Dr. B.D.Subudhi, Professor and Head, Department Of Electrical Engineering, Dr.Sandeep Gosh, Dr. J.K. Satpathy, Dr. P.K. Sahu, Dr. Susmita Das and Dr. Dipti Patra** for providing a solid foundation for my studies and research thereafter. They have been great sources of inspiration to me and I thank them from the bottom of my heart.

I express special thanks to all my friends and colleagues who have inspired me a lot during the course of research work.

Finally, I would like to thank my family for their continued support and encouragement. They rendered me enormous support during the course of this project work.

Sheeja K.L.

Contents

ABSTRACT

One of the major practical problems in digital communication systems is channel distortion which causes errors due to intersymbol interference. Since the source signal is in general broadband, the various frequency components experience different steady state amplitude and phase changes as they pass through the channel, causing distortion in the received message. This distortion translates into errors in the received sequence. Our problem as communication engineers is to restore the transmitted sequence or, equivalently, to identify the inverse of the channel, given the observed sequence at the channel output. This task is accomplished by adaptive equalizers. Typically, adaptive equalizers used in digital communications require an initial training period, during which a known data sequence is transmitted. A replica of this sequence is made available at the receiver in proper synchronism with the transmitter, thereby making it possible for adjustments to be made to the equalizer coefficients in accordance with the adaptive filtering algorithm employed in the equalizer design. When the training is completed, the equalizer is switched to its decision directed mode. Decision feedback equalizers are used extensively in practical communication systems. They are more powerful than linear equalizers especially for severe inter-symbol interference (ISI) channels without as much noise enhancement as the linear equalizers. This thesis addresses the problem of adaptive channel equalization in environments where the interfering noise exhibits Gaussian behavior. In this thesis, radial basis function (RBF) network is used to implement DFE. Advantages and problems of this system are discussed and its results are then compared with DFE using multi layer perceptron net (MLP).Results indicate that the implemented system outperforms both the least-mean square(LMS) algorithm and MLP, given the same signal-to-noise ratio as it offers minimum mean square error. The learning rate of the implemented system is also faster than both LMS and the multilayered case.

Keywords: - Non-linear equalization, neural networks, radial basis function, LMS, decision feedback equalizers, ISI channels.

Acronyms

ISI	Inter Symbol Interference
LMS	Least Mean Square
RLS	Recursive Least Square
MLP	Multi Layer Perceptron
MMSE	Minimum mean square error
MSE	Mean Square Error
BER	Bit Error Rate
LAN	Local Area Network
DSP	Digital Signal Processing
RBF	Radial Basis Function
FIR	Finite Impulse Response
BP	Back Propagation
BPSK	Binary Phase Shift Keying
MLSE	Maximum Likelihood Sequence Estimator
TE	Transversal Equalizer
DFE	Decision Feedback Equalizer
FFMLP	Feed Forward Multi Layer Perceptron
MAP	Maximum a Posteriori
IIR	Infinite Impulse Response
ANN	Artificial Neural Network

List of Figures

CHAPTER 1

INTRODUCTION

1. INTRODUCTION

1.1 INTRODUCTION

Adaptive signal processing plays a crucial role in many modern communication systems. Adaptive equalization is a particular example of adaptive signal processing. It is an important technique to combat distortion and interference in communication links. The conventional approach to communication channel equalization is based on adaptive linear system theory. Channel equalization is an important subsystem in a communication receiver. Equalization is a technique used to remove inter-symbol interference (ISI) produced due to the limited bandwidth of the transmission channel [1]. When the channel is band limited, symbols transmitted through will be dispersed. This causes previous symbols to interfere with the next symbols, yielding the ISI. Also, multipath reception in wireless communications causes ISI at the receiver. Thus, equalizers are used to make the frequency response of the combined channel-equalizer system flat.

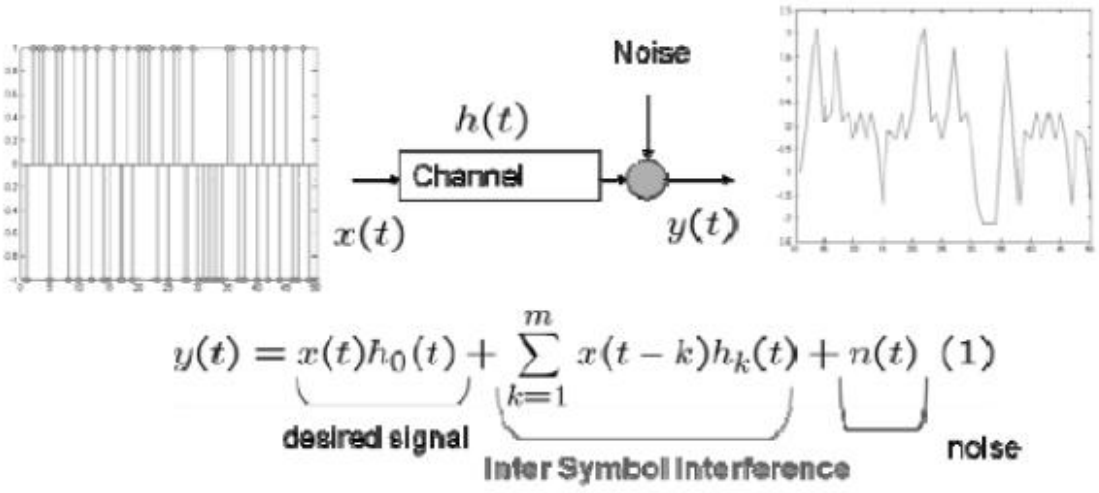


Fig.1.1 Inter Symbol Interference

The purpose of an equalizer is to reduce the ISI as much as possible to maximize the probability of correct decisions.

1.2 Adaptive equalizer classification

This section provides adaptive equalizer classification as presented in Figure 2.4. In general the family of adaptive equalizers is either supervised or unsupervised. The channel distortions introduced into the signal can be conveniently removed by sending a training or pilot signal periodically during the transmission of information. A replica of this pilot signal is available at the receiver which uses this to update its parameters during the training period. These types are *supervised equalizers*. However, in certain communication systems like digital television and digital radio, there is hardly any scope for the use of a training signal. In such situations the equalizer needs some form of unsupervised or self-recovery method to update its parameters so as to provide near optimal performance. These are called *blind equalizers*. This thesis investigates *supervised equalizers* in general.

1.3 Types of Equalizers

Two classes of equalizers are known: linear and non-linear equalizers.

1.3.1 Linear Equalizer

In a Linear Equalizer, the current and the past values of the received signal are linearly weighted by equalizer coefficients and summed to produce the output, using the relation below.

$$C(z) = \sum_{k} c_k z^{-k}$$

a) Mean-Square Error equalizer– Such type of equalizers attempts to minimize the total error between the slicer input and the transmitted data symbol.

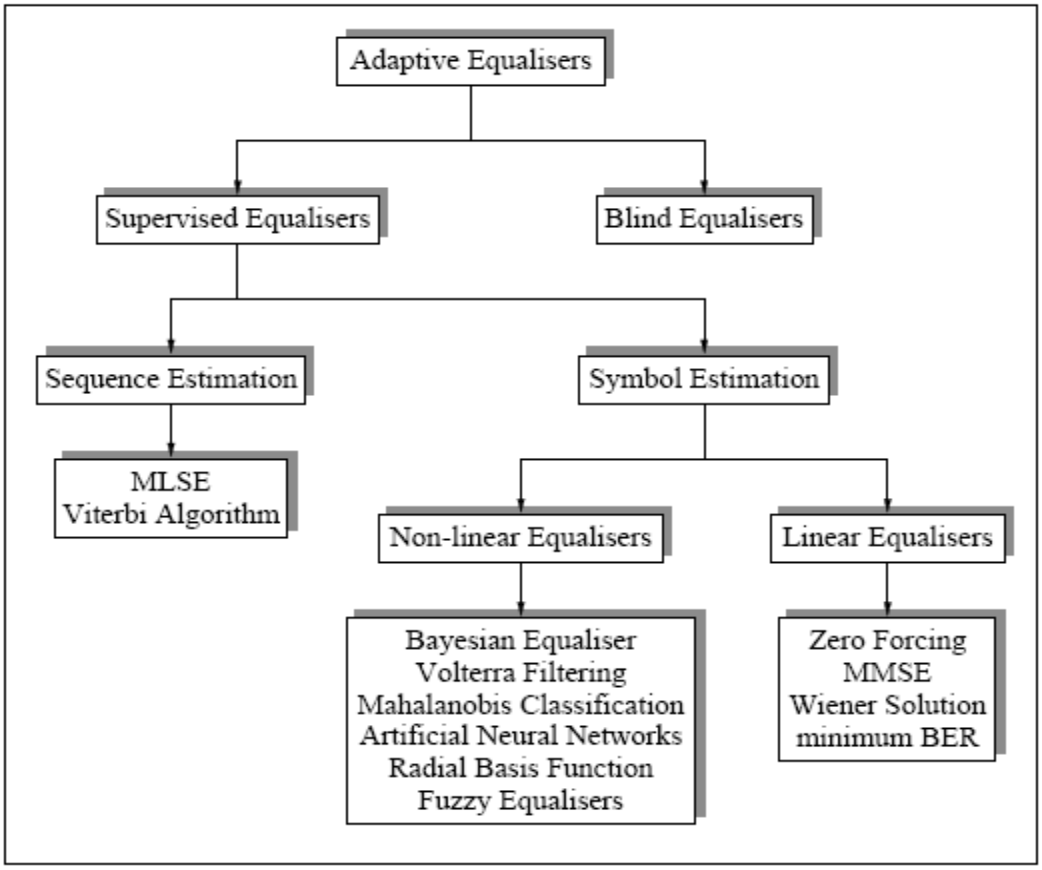


Fig.1.2 Classification of Adaptive Equalizers

1.3.2 Non-linear equalizers

a) Decision Feedback Equalizer-It is a simple nonlinear equalizer which is particularly useful for channel with severe amplitude distortion. It uses decision feedback to cancel the interference from symbols which have already have been detected. The equalized signal is the sum of the outputs of the forward and feedback parts of the equalizer. The forward part is like the linear transversal equalizer. Decisions made on the equalized signal are fed back via a second transversal filter. The basic idea is that if the values of the symbols already detected are known (past decisions are assumed correct), then the ISI contributed by these symbols can be canceled exactly, by subtracting past symbol values with appropriate weighting from the equalizer output.

Since the output of the feedback section of the DFE is a weighted sum of noise-free past decisions, the feedback coefficients play no part in determining the noise power at the equalizer output. However, the DFE can compensate for amplitude distortion without as much noise enhancement as a linear equalizer. The DFE performance is also less sensitive to the sampler phase. An intuitive explanation for these advantages is as follows:

The coefficients of a linear transversal equalizer are selected to force the combined channel and equalizer impulse response to approximate a unit pulse. In a DFE, the ability of the feedback section to cancel the ISI, because of a number of past symbols, allows more freedom in the choice of the coefficients of the forward section. The combined impulse response of the channel and the forward section may have nonzero samples following the main pulse. That is, the forward section of a DFE need not approximate the inverse of the channel characteristics, and *so* avoids excessive noise enhancement and sensitivity to sampler phase.

When a particular incorrect decision is fed back, the DFE output reflects this error during the next few symbols as the incorrect decision traverses the feedback delay line. Thus there is a greater likelihood of more incorrect decisions following the first one, i.e., error propagation. Fortunately, the error propagation in a DFE is not catastrophic. On typical channels, errors occur in short bursts that degrade performance only slightly.

In a Decision Feedback Equalizer Architecture (DFE), shown in figure 1.2 consists of a linear feedforward filter (FFF) and a feedback filter (FBF). The FFF suppresses the contribution of the pre-cursor ISI, namely the interference caused by the symbols transmitted after the symbol of interest. The FBF cancels the post-cursor ISI by subtracting a weighted linear combination of

the previous symbol decisions, assumed to be correct. The result is then applied to a threshold device to determine the symbol of interest. The FFF enhances the noise, but the noise gain is not as severe as in the case of a linear equalizer. Both the forward and feedback filters may be adjusted simultaneously to minimize the Mean Square Error.

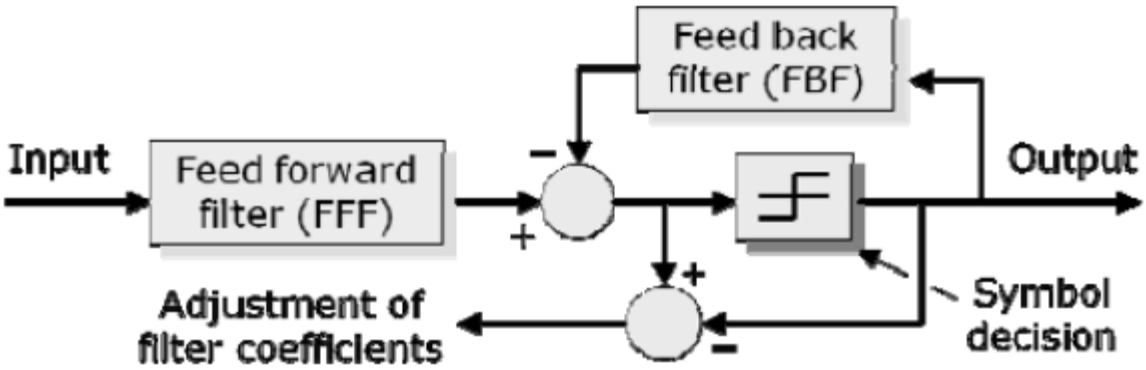


Fig.1.3 Decision Feedback Equalizer

1.3.3 Adaptive Equalizers

This type of equalizers adapts the coefficients to minimize the noise and intersymbol interference (depending on the type of equalizer) at the output.

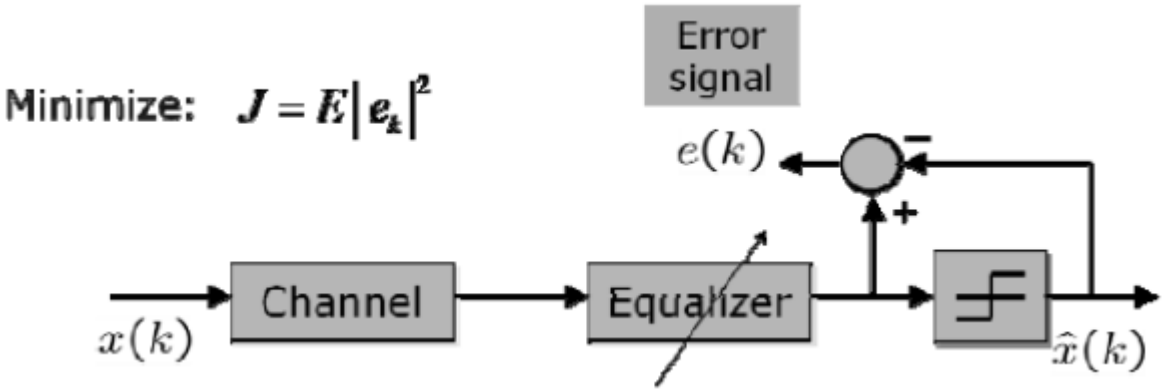


Fig.1.4 Adaptive Equalizers

There are two modes that adaptive equalizers work;

a) Decision Directed Mode: The receiver decisions are used to generate the error signal. Decision directed equalizer adjustment is effective in tracking slow variations in the channel response. However, this approach is not effective during initial acquisition.

b) Training Mode: To make equalizer suitable in the initial acquisition duration, a training signal is needed. In this mode of operation, the transmitter generates a data symbol sequence known to the receiver. Once an agreed time has elapsed, the slicer output is used as a training signal and the actual data transmission begins.

1.4 Background

This thesis discusses the development of adaptive channel equalizers for communication channels using MLP and RBF networks. In order to establish the context and motivation for this research work undertaken clearly and coherently, it is necessary to discuss the background behind the developments in DFE design.

The idea of using previous decisions to cope with the ISI problem was first introduced in 1967 by Austin [15], only two years after the development of the digital (adaptive) linear equalizer by Lucky [25]. The decision feedback receiver that minimizes the MSE between the input to the threshold device and the transmitted symbol was first obtained by Monsen [27]. In [26], the joint optimization of the transmitter and receiver, to maximize the output signal-to-noise ratio (SNR) of a zero-forcing DFE, was considered by Price. The jointly optimum MMSE transmitter and receiver were obtained by Salz [28]. Using linear space geometric arguments Messerschmitt [29] showed the equivalence of zero-forcing (ZF) decision feedback to MMSE prediction of a random process (and also the equivalence of a linear ZF receiver to linear interpolation of a random process), and thus provided simple derivations of the optimum filters and

conditions for their existence. In [30], Belfiore and Park introduced a new DFE structure, called the noise predictive DFE and showed its equivalence to the conventional DFE for infinite-length filters. Infinite-length results on the MMSE-DFE were extended to the finite-length case in [31]. Thus, developments in DFE design have continued over 30 years since its inception.

1.5 MOTIVATION FOR WORK

Artificial Neural Nets (ANNs) are able to perform complex non linear classification problems and hence they can be used as equalizers. Most ANNs use the mean square error (MSE) as the cost function to be minimized by the network. Problems encountered using ANNs in equalization are the slow rate of convergence and the possibility that the net does not reach the true minimum mean square error MSE[7].In this case, the net will not be able to optimize its parameters to the least MSE. Two ANN models are used in this thesis, namely MLP and RBF nets.

Several approaches using ANNs in equalization have been proposed in the last few years. Kirkland in 1992 used feed forward ANNs in equalizing a multipath fading channel [8]. In the same year, Peng modified the activation function of the MLP to be suitable for phase-amplitude modulation (PAM) and quadrature-amplitude modulation (QAM) schemes [9]. In 1994, Kechriotis used recurrent ANNs in equalizing different linear and non-linear channels [10]. Chang, in the same year, introduced a neural-based DFE to equalize indoor radio channel [11]. He also used a wavelet ANN trained with recursive least squares (RLS) algorithm to equalize a non linear channel. Al-Mashouq used a feed forward NN to combine both equalization and decoding at the receiver [12]. This method performed better than the cascaded equalizer-decoder pair. Mulgrew investigated the implementation of DFEs using RBF nets in 1996[3].

In 1997 a new algorithm for training recurrent NN was proposed [6]. It was called the discriminative least square (DLS) and it was faster to converge than the RLS and LMS algorithms.

In this thesis, an RBF net is used as a DFE. The paper discusses architectures of the DFE and the RBF net. Then the use of RBF net to implement a DFE is presented. Simulation results are then discussed. Finally, conclusions and suggestions for future work are presented.

1.6 THESIS LAYOUT

The complete layout of the present thesis proceeds as follows:

Chapter 1 gives an introduction to channel equalization and their classification. It deals with linear and non-linear equalizers especially decision feedback equalizers.DFE architecture is also discussed. It includes the background behind the developments in DFE design and motivation behind undertaking this thesis work.

Chapter 2 deals with the basic principles of channel equalization .Herein it introduces a base band model of a digital communication system. Also, multipath propagation and different types of channels are discussed. It deals with ISI and gives a brief over view on linear transversal equalizer, decision feedback equalizer structures and various other non linear equalizer structures.

Chapter 3 describes the three gradient based training methods such as the LMS algorithm, BP algorithm and RBF networks used to train the equalizer parameters. It deals with a single neuron structure, different types of non-linear activation function and multilayer perceptron architecture. In case of RBF, multidimensional interpolation method is discussed in detail. Lastly, RBF

networks, MLP and LMS is compared theoretically which is proved through simulations in the succeeding chapters.

Chapter 4 deals with linear equalization using LMS algorithm. Here the system model (FIR model of the channel) and equalizer model is discussed. It gives a detailed structure of the transversal filter and adaptive weight control mechanism. It deals with the signal flow graph representation of the LMS algorithm. Lastly the simulation is carried out to evaluate the response of the adaptive equalizer using the LMS algorithm to changes in the step size parameter.

Chapter 5 deals with decision feedback equalization using LMS and MLP. Here a base band transmission system is introduced. Also, DFE structure and perceptron based DFE architecture used for simulation is discussed. The convergence characteristics and BER performance for both MLP-DFE and LMS-DFE is simulated with{(4,1)DFE}and without{(5,0)DFE} feedback signals and the findings are compared in this chapter.

Chapter 6 deals with decision feedback equalization using MLP and RBF net. Here a RBF net is used as a DFE. Two proakis channels are used for simulation; one is a linear channel while the other is a severe ISI channel with a deep frequency null. (5,0) and (4,1) DFEs are implemented using both MLP and RBF nets and the BER performance is simulated. The results are then compared for both channels. In case of (4,1) DFE , two cases were simulated ,i.e., when correct symbols and detected symbols are used as feedback signals. Convergence of both MLP and RBF is simulated and their results are compared.

Chapter 7 deals with the conclusion of the investigation made in the thesis. This chapter also suggests some future research related to the topic

1.7 THESIS CONTRIBUTIONS

The major contribution of the thesis is outlined below:

• Linear equalizers using LMS algorithm are pre-dominantly controlled by its slower modes of convergence while the fast modes are hardly visible. It is due to the fact that the rate of convergence of the adaptive equalizers is highly dependent on the step-size parameter.

• Linear equalizers are not good for severe ISI channels; hence a DFE is implemented using both MLP and RBF nets as DFEs are more powerful than Linear equalizers without as much noise enhancement.

• MLP based DFE provides better BER performance in comparison to LMS based DFE when the level of additive noise is high in cases of both with and without feedback signals. Also, MLP based DFE converges faster than that of LMS based DFE.

• RBF based DFE performs better than that of MLP based DFE at high SNR. Moreover, RBF converges faster than that of MLP in the training mode but needs more computational time in the decision directed mode, because of its large number of neurons compared with the MLP.

• DFE performs better when correct symbols are used as feedback signals rather than when detected symbols are used as feedback signals.

Chapter 2

Basic Principles of Channel Equalization

2. Basic Principles of Channel Equalization

2.1 INTRODUCTION

In an ideal communication channel, the received information is identical to that transmitted. However, this is not the case for real communication channels, where signal distortions take place. A channel can interfere with the transmitted data through three types of distorting effects: power degradation and fades, multi-path time dispersions and background thermal noise [1]. Equalization is the process of recovering the data sequence from the corrupted channel samples. A typical baseband model of a digital communication system is depicted in Fig2.1, where an equalizer is incorporated within the receiver [32].

The equalization approaches investigated in this thesis are applied to a BPSK (binary phase shift keying) baseband communication system. Each of the transmitted data belongs to a binary and 180 out of phase alphabet $\{-1,+1\}$.

Within this chapter channel baseband models are explained. A transversal equalizer structure is also examined [33].

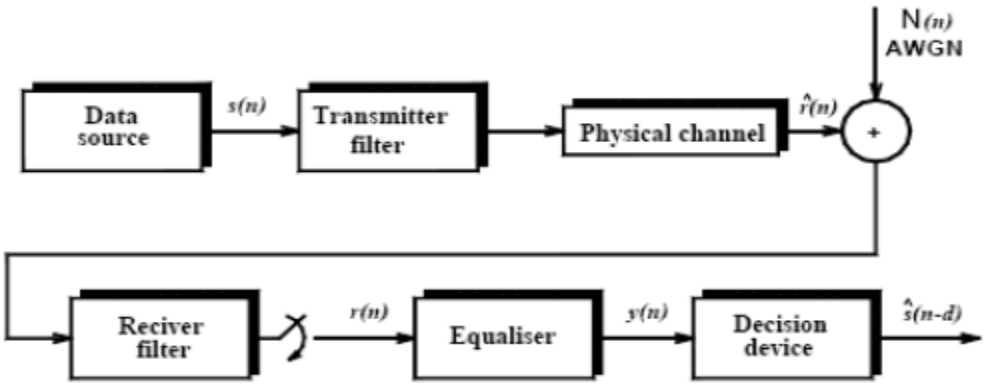


Fig.2.1 Baseband model of a Digital Communication System

The block diagram of baseband model of a digital communication system (DCS) is depicted in Figure 2.1. Communication systems are studied in the base band frequency to avoid the complexity associated with the analysis of various subsystems within a DCS. The data source constitutes the signal generation

system that originates the information to be transmitted. The efficient use of the available bandwidth is achieved through the transmitter filter, also called the modulating filter. The channel is the medium through which information propagates from the transmitter to the receiver. At the receiver the signal is first demodulated to recover the baseband transmitted signal. This demodulated signal is processed by the receiver filter, also called the receiver demodulating filter, which should be ideally matched to the transmitter filter and channel. The equalizer in the receiver removes the distortion introduced due to the channel impairments. The decision device provides the estimate of the transmitted signal. During transmission of high speed data over a band-limited channel, its frequency response is usually not known with sufficient precision to design an optimum match or matched filter. The equalizer is therefore, should be adaptive in nature to take care of the variations in the characteristics of the channel.

2.1.1 MULTIPATH PROPAGATION

Within telecommunication channels multiple paths of propagation commonly occur. In practical terms this is equivalent to transmitting the same signal through a number of separate channels, each having a different attenuation and delay [1] .Consider an open-air radio transmission channel that has three propagation paths, as illustrated in Fig2.2 [16]. These could be direct, earth bound and sky bound.Fig2.1b describes how a receiver picks up the transmitted data. The direct signal is received first whilst the earth and sky bound are delayed. All three of the signals are attenuated with the sky path suffering the most.

Multipath interference between consecutively transmitted signals will take place if one signal is received whilst the previous signal is still being detected [1]. In Fig2.1 this would occur if the symbol transmission rate is greater than

$1/\tau$ where, τ represents transmission delay. Because bandwidth efficiency leads to high data rates, multi-path interference commonly occurs.

Channel models are used to describe the channel distorting effects and are given as a summation of weighted time delayed channel inputs d (n- i).

$$H(z) = \sum_{i=0}^{m} d(n-i)z^{-i} = d(n) + d(n-1)z^{-1} + d(n-2)z^{-2} + \ldots\ldots\ldots \quad (2.1)$$

The transfer function of a multi-path channel is given in (2.1). The model coefficients d (n- i) describe the strength of each multipath signal.

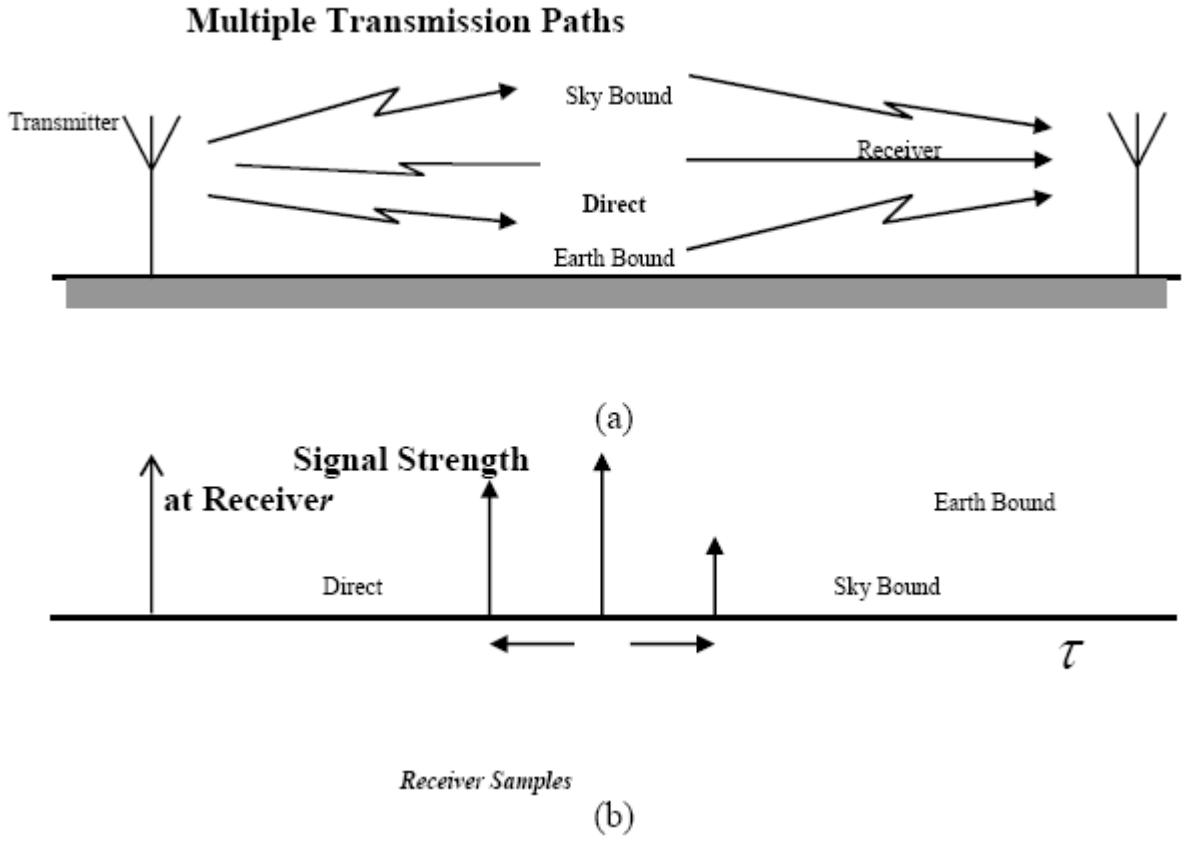


Figure 2.2: Impulse Response of a transmitted signal in a channel which has 3 modes of propagation, (a) The signal transmitted paths, (b) The received samples

2.2 MINIMUM AND NONMINIMUM PHASE CHANNELS

When all the roots of the H (z) lie within the unit circle, the channel are termed minimum phase. The inverse of a minimum phase [34] channel is convergent, illustrated by (2.2):

$$\begin{aligned} H(z) &= 1.0 + 0.5\, z^{-1} \quad \frac{1}{H(z)} \\ \frac{1}{H(z)} &= \frac{1}{1.0 + 0.5z^{-1}} \\ &= \sum_{i=0}^{\infty} \left(\frac{-1}{2}\right)^{i} z^{-i} \\ &= 1 - 0.5z^{-1} + 0.25z^{-2} - 0.125z^{-3} + \ldots\ldots \end{aligned} \qquad (2.2)$$

whereas the inverse of non-minimum phase channels are not convergent, as shown in (2.3)

$$\begin{aligned} H(z) &= 0.5 + 1.0z^{-1} \\ \frac{1}{H(z)} &= \frac{z}{1.0 + 0.5z} \\ &= z.\left[\sum_{i=0}^{\infty} \left(-\frac{1}{2}\right)^{i} z^{-i}\right] \\ &= z.\left[1 - 0.5z + 0.25z^{2} - 0.125z^{3}\right] \end{aligned} \qquad (2.3)$$

Since equalizers are designed to invert the channel distortion process they will in effect model the channel inverse. The minimum phase channel has a linear inverse model therefore a linear equalization solution exists. However, limiting the inverse model to m-dimensions will approximate the solution and it has been

shown that non-linear solutions can provide a superior inverse model in the same dimension [32] .

A linear inverse of a non-minimum phase channel does not exist without incorporating time delays. A time delay creates a convergent series for a non-minimum phase model, where longer delays are necessary to provide a reasonable equalizer. (2.4) describes a non-minimum phase channel with a single delay inverse and a four sample delay inverse. The latter of these is the more suitable form for a linear filter.

$$\left.\begin{aligned} &H(z) = 0.5 + 1.0z^{-1} \\ &z^{-1}\frac{1}{H(z)} = \frac{1}{1+0.5z} = 1 - 0.5z + 0.25z^2 - 0.125z^3 + ...(\,noncausal) \\ &z^{-4}\frac{1}{H(z)} = z^{-3} - 0.5z^{-2} + 0.25z^{-1} - 0.125z + ...(\,truncatedandcausal) \end{aligned}\right\} \quad (2.4)$$

The three-tap non-minimum phase channel $H(z) = 0.3410 + 0.8760\,z^{-1} + 0.3410\,z^{-2}$ is used throughout this thesis for simulation purposes. A channel delay, D, is included to assist in the classification so that the desired output becomes *u* (*n-D*).

2.3 INTERSYMBOL INTERFERENCE

Inter-symbol interference (ISI) has already been described as the overlapping of the transmitted data [1]. It is difficult to recover the original data from one channel sample dimension because there is no statistical information about the multipath propagation. Increasing the dimensionality of the channel output vector helps characterize the multipath propagation. This has the affect of not only increasing the number of symbols but also increases the Euclidean distance between the output classes.

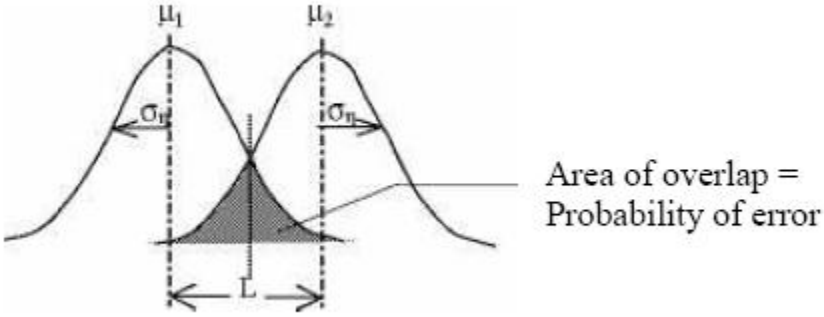


Fig.2.3 Interaction between two neighboring symbols

When additive Gaussian noise, η is present within the channel, the input sample will form Gaussian clusters around the symbol centers. These symbol clusters can be characterized by a probability density function (pdf) with a noise variance σ_η^2, where the noise can cause the symbol clusters to interfere. Once this occurs, equalization filtering will become inadequate to classify all of the input samples. Error control coding schemes can be employed in such cases but these often require extra bandwidth [35].

2.3.1 SYMBOL OVERLAP

The expected number of errors can be calculated by considering the amount of symbol interaction, assuming Gaussian noise. Taking any two neighboring symbols, the cumulative distribution function (CDF) can be used to describe the overlap between the two noise characteristics. The overlap is directly related to the probability of error between the two symbols and if these two symbols belong to opposing classes, a class error will occur. Fig2.3 shows two Gaussian functions that could represent two symbols noise distributions. The Euclidean distance, L, between symbol centers and the noise variance, σ^2, can be used in the cumulative distribution function of (2.5) to calculate the area of overlap between the two symbol noise distributions and therefore the probability of error, as in (2.6).

$$CDF(x) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{x^2}{2\sigma^2} \right) dx \qquad (2.5)$$

$$P(e) = 2CDF\left(\frac{L}{2} \right) \qquad (2.6)$$

Since each channel symbol is equally likely to occur [33], the probability of unrecoverable errors occurring in the equalization space can be calculated using the sum of all the CDF overlap between each opposing class symbol. The probability of error is more commonly described as the BER. (2.7) describes the BER based upon the Gaussian noise overlap, where N_{SP} is the number of symbols in the positive class, N_m is the number of symbols in the negative class and Δ_i , is the distance between the ith positive symbol and its closest neighboring symbol in the negative class.

$$BER(\sigma_n) = \log\left[\frac{2}{N_{sp} + N_m} \sum_{i=1}^{N_{sp}} CDF\left(\frac{\Delta i}{2\sigma_n} \right) \right] \qquad (2.7)$$

2.4 CHANNEL EQUALIZATION

The optimal BER equalization performance is obtained using a maximum likelihood sequence estimator (MLSE) on the entire transmitted data sequence [36].A more practical MLSE would operate on smaller data sequences but these can still be computationally expensive, they also have problems tracking time-varying channels and can only produce sequences of outputs with a significant time delay. Another equalization approach implements a symbol-by-symbol detection procedure and is based upon adaptive filters [1]. The symbol-by-

symbol approach to equalization applies the channel output samples to a decision classifier that separates the symbol into their respective classes. Two types of symbol-by-symbol equalizers are examined in this thesis, the transversal equalizer (TE) and decision feedback equalizer (DFE). Traditionally these equalizers have been designed using linear filters, LTE and LDFE, with a simple FIR structure. The ideal equalizer will model the inverse of the channel model but this does not take into account the effect of noise within the channel.

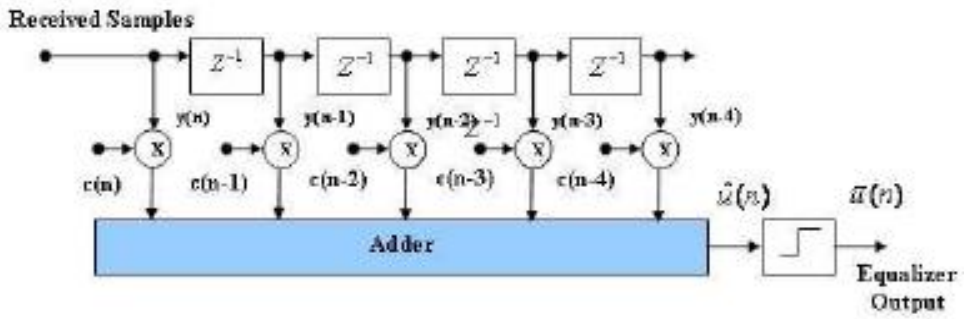


Fig.2.4 Linear Transversal Equalizer

2.4.1 TRANSVERSAL EQUALIZER

The transversal equalizer uses a time-delay vector, Y (n) (2.8), of channel output samples to determine the symbol class. The $\{m\}$ TE notation used to represent the transversal equalizer specifies m inputs. The equalizer filter output will be classified through a threshold activation device (Fig2.4) so that the equalizer decision will belong to one of the BPSK states $\bar{u}(n) \in \{-1,+1\}$.

$$Y(n) = [y(n), y(n-1), ..., y(n-(m-1))] \qquad (2.8)$$

Considering the inverse of the channel $H(z) = 1.0 + 0.5z^{-1}$ that was given in (2.3), this is an infinitely long convergent linear series: $\frac{1}{H(z)} = \sum_{i=1}^{m} \left(-1/2\right)^i z^{-i}$. Each coefficient of this inverse model can be used in a linear equalizer as a FIR tap-weight. Each tap-dimension will improve the accuracy; however, high input

dimensions leave the equalizer susceptible to noisy samples. If a noisy sample is received, this will remain within the filter affecting the output from each equalizer tap. Rather than designing a linear equalizer, a non-linear filter can be used to provide the desired performance that has a shorter input dimension; this will reduce the sensitivity to noise.

2.4.2 Decision Feedback Equalizer

A basic structure of the decision feedback equalizer (DFE) is shown in Fig2.5.The DFE consists of a transversal feed forward and feedback filter. In the case when the communication channel causes severe ISI distortion, the LTE could not be provide satisfactory performance. Instead, a DFE is required. The DFE uses past corrected samples, w (n), from a decision device to the feedback filter and combine with the feed forward filter. In effect, the function of the feedback filter is to subtract the ISI produced by previously detected symbols from the estimates of future samples.

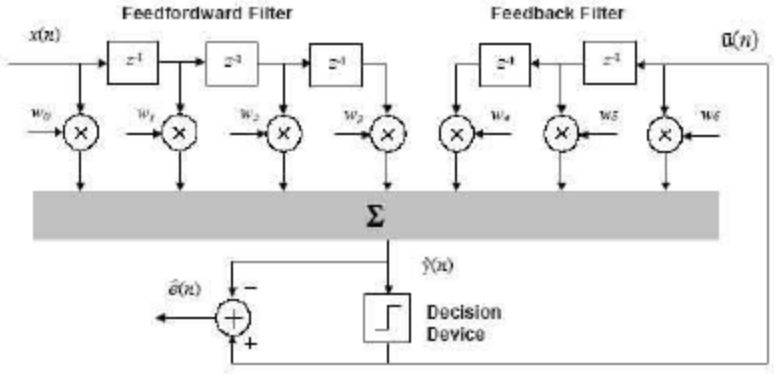


Fig.2.5 Decision Feedback Equalizer

Consider that the DFE is updated with a recursive algorithm; the feed forward filter weights and feedback filter weights can be jointly adapted by the LMS algorithm on a common error signal $\hat{e}(n)$ as shown in (2.9).

$$W(n+1) = W(n) + \mu\hat{e}(n)V(n) \qquad (2.9)$$

Where $\hat{e}(n) = u(n) - y(n)$

and $V(n) = [x(n), x(n-1),\ldots.., x(n-k1-1), u(n-k2-l),\ldots.u(n)]^T$

The feed forward and feedback filter weight vectors are written in a joint vector as $W(n) = [w_0(n), w_1(n),\ldots.., w_{k1+k2-1}(n)]^T$. k1 and k2 represent the feed forward and feedback filter tap lengths respectively. Suppose that the decision device causes an error in estimating the symbol u (n). This error can propagate into subsequent symbols until the future input samples compensate for the error. This is called the error propagation which will cause a burst of errors. The detrimental potential of error propagation is the most serious drawback for decision feedback equalization. Traditionally, the DFE is described as being a non-linear equalizer because the decision device is non-linear. However, the DFE structure is still a linear combiner and the adaptation loop is also linear. It has therefore been described as a linear equalizer structure.

2.4.3 NON-LINEAR EQUALIZER STRUCTURES

Recently there has been interest into the application of non-linear architectures to the equalization problem, with the aim of enhancing the noise performance as well as the channel non-linearity. Both LTE and DFE architectures can benefit from the implementation of these structures, by showing an enhanced Bit Error Rate (BER) performance when compared to conventional linear architectures .Three particular types of non-linearity have been investigated, the Gaussian radial basis function (RBF), the feed forward multilayer perceptron (FFMLP), and the Volterra Kernel.

The Gaussian RBF equalizer has been suggested as a solution to the fast fading time varying mobile telecommunications systems, where its adaptation to the non-stationary channel model has been shown to surpass the performance of a more conventional maximum likelihood sequence estimator MLSE. The RBF

model also is surprisingly parsimonious when compared to the MLSE. However, as the dimensionality of the input increases, the number of indicated kernels also increases. If the kernel centers are not identified with a high degree of accuracy the system can be over specified.

The Volterra kernel (third order) has also been utilized in satellite communication channels, and as such it can be trained utilizing a least squares training algorithm. However like the RBF kernel the Volterra series suffers from the curse of dimensionality caused by the proliferation of the cross coefficients. This problem can be alleviated by a careful choice of the desired polynomial, which will result in the polynomial structure being both parsimonious and trainable using the Support Vector (SV) approach. The FFMLP was the first multilayer neural network structure to be implemented after a method of training was discovered. Work by Siu (1990) has shown the feasibility of using these non-linear structures to equalize time delayed non-minimum phase channels; however, as it seems with all non-linear architectures, training difficulties tend to limit their effectiveness. It has been shown that the non-linear boundaries could be close to the optimal maximum a posteriori (MAP) boundary, which is formed by utilizing a Gaussian RBF network with centers at all of the possible signal is formed by utilizing a Gaussian RBF network with centers at all of the possible signal centers . It is possible to train a FFMLP with fewer processing units than that generated by the MAP criterion, and thus have a more parsimonious structure. There has, however, been a tendency to train to linear solutions that do not truly reject the non-linear nature of the decision surface. The primary reason for this was that the gradient descent training schemes employed tend to cause premature convergence to local minima, as well as algorithmic instability, due primarily to the topology of the error surface. It has been shown that gradient descent can fail even when the FFMLP structure itself is sufficient to deal with the problem. This chapter discussed the background of channel equalization and highlights some of the most common equalizer structures, the LTE and the DFE.

Both the linear and non-linear methods have been discussed with the aim of highlighting the necessity of the non-linear architecture, even though we have used a linear equalizer as the test problem.

2.5 SUMMARY

This chapter explains the needs and different methods of channel equalization. The natures of minimum and non-minimum phase channels are described. It is seen that the equalizer dimension is large for non-minimum channels. Various interferences in communication channels are addressed. Multipath interference is explained briefly. A transversal equalizer and decision feedback equalizer is briefly explained. Finally the nonlinear equalizer structures are explained briefly.

CHAPTER 3

Different adaptive algorithms

3. Different adaptive algorithms

3.1. THE LMS ALGORITHM

In case of LMS algorithm the cost function $J_{LMS}(N)$ is given by

$$J_{LMS}(n) = \frac{1}{2}e^2(n) \qquad (3.1)$$

This cost function can be thought of as an instantaneous estimate of the MSE cost function, as $J_{MSE}(n) = EJ_{LMS}(n)$. Although it might not appear to be useful, the resulting algorithm obtained when $J_{LMS}(N)$ is used for J (n) given by (3.2).

$$W(n+1) = W(n) - \mu(n)\frac{\delta J(n)}{\delta W(n)} \qquad (3.2)$$

is extremely useful for practical applications. Taking derivatives of $J_{LMS}(n)$ with respect to the elements of W (n) and substituting the result into (3.2), we obtain the LMS adaptive algorithm given by

$$W(n+1) = W(n) + \mu(n)e(n)X(n) \qquad (3.3)$$

Note that this algorithm requires only multiplications and additions to implement. In fact, the number and type of operations needed for the LMS algorithm is nearly the same as that of the FIR filter structure with fixed coefficient values, which is one of the reasons for the algorithm's popularity.

The behavior of the LMS algorithm has been widely studied, and numerous results concerning its adaptation characteristics under different situations have been developed. For now, we indicate its useful behavior by noting that the solution obtained by the LMS algorithm near its convergent point is related to the Wiener solution. In fact, analyses of the LMS algorithm under certain statistical assumptions about the input and desired response signals show that when the Wiener solution $W_{MSE}(n)$ is a fixed vector. Moreover, the average behavior of the LMS algorithm is quite similar to that of the steepest descent algorithm that depends when the Wiener solution $W_{MSE}(n)$ is a fixed

$$\lim_{n\to\infty} E[W(n)] = W_{MSE} \qquad (3.4)$$

vector. Moreover, the average behavior of the LMS algorithm is quite similar to that of the steepest descent algorithm that depends explicitly on the statistics of the input and desired response signals. In effect, the iterative nature of the LMS coefficient updates is a form of time-averaging that smoothes the errors in the instantaneous gradient calculations to obtain a more reasonable estimate of the true gradient. The problem is that gradient descent is a local optimization technique, which is limited because it is unable to converge to the global optimum on a multimodal error surface if the algorithm is not initialized in the basin of attraction of the global optimum.

Several modifications exist for gradient based algorithms in attempt to enable them to overcome local optima. One approach is to simply add noise or a momentum term [30] to the gradient computation of the gradient descent algorithm to enable it to be more likely to escape from a local minimum. This approach is only likely to be successful when the error surface is relatively smooth with min or local minima, or some information can be inferred about the topology of the surface such that the additional gradient parameters can be assigned accordingly.

Other approaches attempt to transform the error surface to eliminate or diminish the presence of local minima [37], which would ideally result in a unimodal error surface.

The problem with these approaches is that the resulting minimum transformed error used to update the adaptive filter can be biased from the true minimum output error and the algorithm may not be able to converge to the desired minimum error condition. These algorithms also tend to be complex, slow to converge, and may not be guaranteed to emerge from a local minimum. Some

work has been done with regard to removing the bias of equation error LMS [37,38] and Stieglitz-McBride [39] adaptive IIR filters, which add further complexity with varying degrees of success.

Another approach [40] attempts to locate the global optimum by running several LMS algorithms in parallel, initialized with different initial co-efficient. The notion is that a larger, concurrent sampling of the error surface will increase the likelihood that one process will be initialized in the global optimum valley. This technique does have potential, but it is inefficient and may still suffer the fate of a standard gradient technique in that it will be unable to locate the global optimum if none of the initial estimates is located in the basin of attraction of the global optimum. By using a similar congregational scheme, but one in which information is collectively exchanged between estimates and intelligent randomization is introduced, structured stochastic algorithms are able to hill-climb out of local minima.

This enables the algorithms to achieve better, more consistent results using a fewer number of total estimates. These types of algorithms provide the framework for the algorithms discussed in the following sections.

3.2. ARTIFICIAL NEURAL NETWORK (ANN)

Artificial neural network (ANN) takes their name from the network of nerve cells in the brain. Recently, ANN has been found to be an important technique for classification and optimization problem [8, 10]. McCulloch and Pitts have developed the neural networks for different computing machines. There are extensive applications of ANN in the field of channel equalization, estimation of parameters of nonlinear systems, pattern recognition, etc. ANN is capable of performing nonlinear mapping between the input and output space due to its large parallel interconnection between different layers and the nonlinear processing characteristics. An artificial neuron basically consists of a computing

element that performs the weighted sum of the input signal and the connecting weight. The sum is added with the bias or threshold and the resultant signal is then passed through a non-linear function of sigmoid or hyperbolic tangent type. Each neuron is associated with three parameters whose learning can be adjusted; these are the connecting weights, the bias and the slope of the nonlinear function. For the structural point of view a NN may be single layer or it may be multilayer. In multilayer structure, there is one or many artificial neurons in each layer and for a practical case there may be a number of layers. Each neuron of the one layer is connected to each and every neuron of the next layer.

The learning of the NN may be supervised in the presence of the desired signal or it may be unsupervised when the desired signal is not accessible. Rumelhart developed the Back propagation algorithm, which is central to much work on supervised learning in multilayer NN. A feed forward structure with input, output, hidden layers and nonlinear sigmoid functions are used in this type of network. In recent years many different types of learning algorithm using the incremental back propagation algorithm, evolutionary learning using the nearest neighbor MLP and a fast learning algorithm based on the layer-by-layer optimization procedure are suggested in literature. In case of unsupervised learning the input vectors are classified into different clusters such that elements of a cluster are similar to each other in some sense. The method is called competitive learning, because during learning sets of hidden units compete with each other to become active and perform the weight change. The winning unit increases its weights on those links with high input values and decreases them on those with low input values. This process allows the winning unit to be selective to some input values. Different types of NNs and their learning algorithms are discussed below.

3.2.1 SINGLE NEURON STRUCTURE

The basic structure of an artificial neuron is presented in Fig. 3.1. The operation in a neuron involves the computation of the weighted sum of inputs and

threshold. The resultant signal is then passed through a nonlinear activation function. This is also called as a preceptor, which is built around a nonlinear neuron; whereas the LMS algorithm described in the preceding sections is built around a linear neuron. The output of the neuron may be represented as,

$$y(n) = \phi[\sum_{j=1}^{N} w_j(n)x_j(n) + \alpha(n)] \qquad (3.5)$$

where $\alpha(n)$ is the threshold to the neurons at the first layer, $w_j(n)$ is the weight associated with the j^{th} input, N is the no. of inputs to the neuron and $\phi(.)$ is the nonlinear activation function.

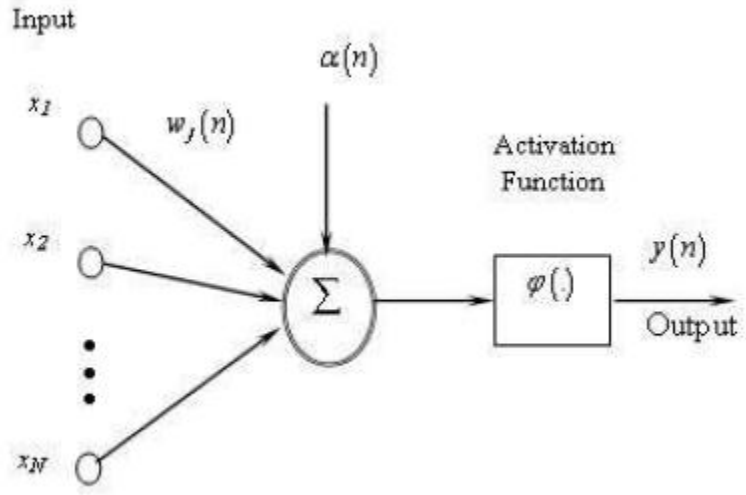


Fig.3.1 Structure of a Single Neuron

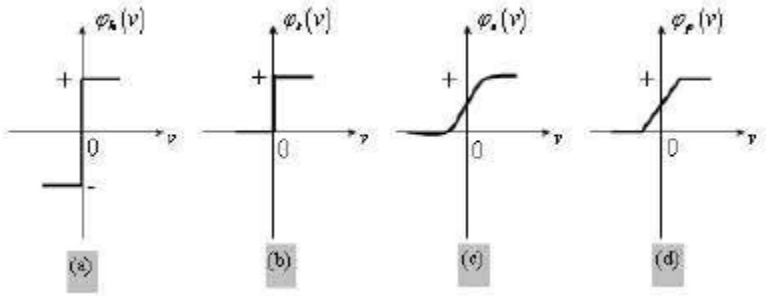


Fig.3.2 Different Types of Non-Linear Activation Function

Different types of nonlinear function are shown in Fig. (3.2). (a) Signum function or hard limiter, (b) Threshold function, (c) Sigmoid function, (d) Piece-wise Linear.

1. Signum Function: For this type of activation function, we have

$$\phi(v) = \begin{cases} 1, \text{ if } v > 0 \\ 0, \text{ if } v=0 \\ -1, \text{ if } v < 0 \end{cases} \qquad (3.6)$$

2. Threshold Function: This function is represented as,

$$\phi(v) = \begin{cases} 1, \text{ if } v \geq 0 \\ 0, \text{ if } v < 0 \end{cases} \qquad (3.7)$$

3. Sigmoid Function: This function is s-shaped, is the most common form of the activation function used in artificial neural network. It is a function that exhibits a graceful balance between linear and nonlinear behavior.

$$\phi(v) = \frac{1}{1+e^{-av}} \qquad (3.8)$$

where v is the input to the sigmoid function and a is the slope of the sigmoid function. For the steady convergence a proper choice of a is required.

4. Piecewise-Linear Function: This function is

$$\phi(v) = \begin{cases} 1, & v \geq +1/2 \\ v, & +1/2 \rangle v \rangle +1/2 \\ 0, & v \leq +1/2 \end{cases} \qquad (3.9)$$

where the amplification factor inside the linear region of operation is assumed to be unity. This can be viewed as an approximation to a nonlinear amplifier.

3.2.2 MULTILAYER PERCEPTRON (MLP)

In the multilayer neural network or multilayer perceptron (MLP), the input signal propagates through the network in a forward direction, on a layer-by-layer basis. This network has been applied successfully to solve some difficult and diverse problems by training in a supervised manner with a highly popular algorithm known as the error back-propagation algorithm [41, 5]. The scheme of MLP using four layers is shown in Fig3.3. x_i (n) represents the input to the network, f_j and f_k represent the output of the two hidden layers and y_l (n) represents the output of the final layer of the neural network. The connecting weights between the input to the first hidden layer, first to second hidden layer and the second hidden layer to the output layers are represented w_{ij} , w_{jk} and w_{kl} by respectively.

If P_1 is the number of neurons in the first layer, each element of the output vector may be calculated as,

$$f_j = \varphi_j \sum_{i=1}^{N} [w_{ij} x_i(n) + \alpha_j], \; j = 1,2,3....P_1 \qquad (3.10)$$

where α_j *the threshold to the neurons at the first layer, N is* is the no. of inputs and $\varphi.$ is the nonlinear activation function. The time index n has been dropped to make the equations simpler. Let P_2 be the number of neurons in the second layer. The output of this layer is represented as, f_k and may be written as

$$f_k = \varphi_k \sum_{j=1}^{P_1} [w_{jk} f_j + \alpha_k], j = 1,2,3....P_2 \qquad (3.11)$$

where, α_k is the threshold to the neurons at the second layer. The output of the final layer can be calculated as

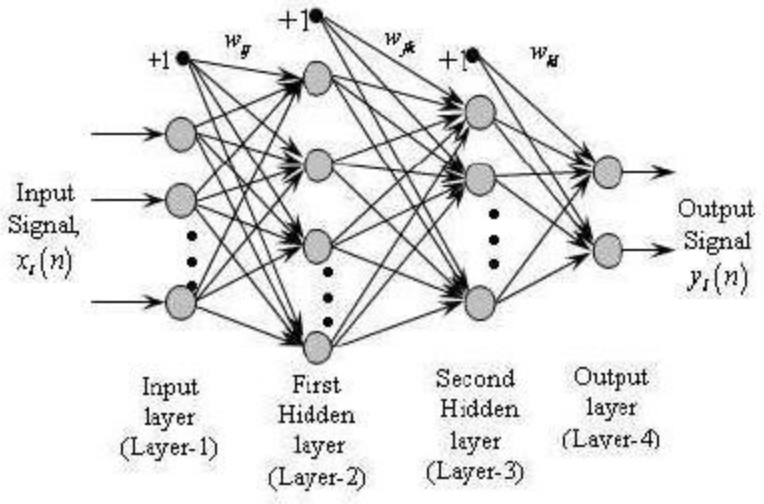


Fig. 3.3 Structure of Multilayer perceptron (MLP)

$$y_l(n) = \varphi_l \sum_{k=1}^{P_2} [w_{kl} f_k + \alpha_l], j = 1,2,3.....P_3 \qquad (3.12)$$

where, α_l is the threshold to the neuron at the final layer and P_3 is the no. of neurons in the output layer. The output of the MLP may be expressed as

$$y_l(n) = \varphi_n[\sum_{k=1}^{P_2} w_{kl}\varphi_k[\sum_{j=1}^{P_1} w_{jk}\varphi_j[\sum_{i=1}^{N} w_{ij}x_i(n) + \alpha_j] + \alpha_k] + \alpha_l] \quad (3.13)$$

3.2.3. Back-propagation (BP) Algorithm

An MLP network with 2-3-2-1 neurons (2, 3, 2 and 1 denote the number of neuron in the input layer, the first hidden layer, the second hidden layer and the output layer respectively) with the back-propagation (BP) learning algorithm, is depicted in Fig.3.4. The parameters of the neural network can be updated in both sequential and batch mode of operation. In BP algorithm, initially the weights and the thresholds are initialized as very small random values. The intermediate and the final outputs of the MLP are calculated by using (3.10), (3.11), and (3.12).

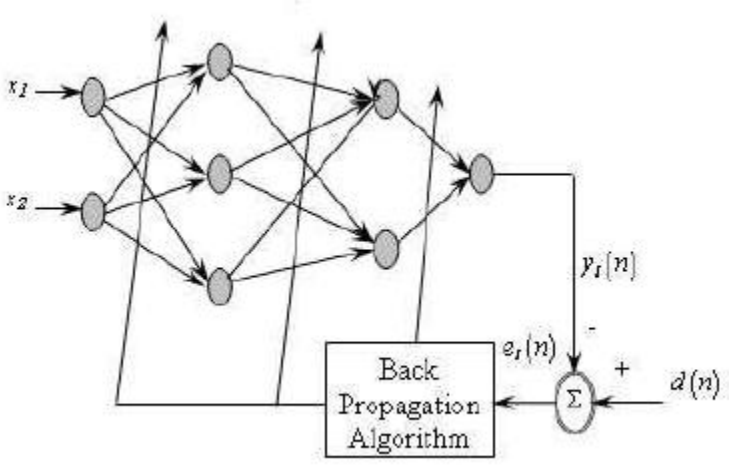


Fig.3.4 Neural Network Training Using BP Algorithm

The final output $y_l(n)$ at the output of neuron l, is compared with the desired output $d(n)$ and the resulting error signal $e_l(n)$ is obtained as

$$e_l(n) = d(n) - y_l(n) \quad (3.14)$$

The instantaneous value of the total error energy is obtained by summing all error signals over all neurons in the output layer, that is

$$\xi(n) = \frac{1}{2}\sum_{l=1}^{P_3} e_l^{\,2}(n) \qquad (3.15)$$

where P_3 is the no. of neurons in the output layer.

This error signal is used to update the weights and thresholds of the hidden layers as well as the output layer. The rejected error components at each of the hidden layers is computed using the errors of the last layer and the connecting weights between the hidden and the last layer and error obtained at this stage is used to update the weights between the input and the hidden layer. The thresholds are also updated in a similar manner as that of the corresponding connecting weights. The weights and the thresholds are updated in an iterative method until the error signal becomes minimum. For measuring the degree of matching, the mean square error (MSE) is taken as a performance measurement.

The updated weights are,

$$w_{kl}(n+1) = w_{kl}(n) + \Delta w_{kl}(n) \qquad (3.16)$$

$$w_{jk}(n+1) = w_{jk}(n) + \Delta w_{jk}(n) \qquad (3.17)$$

$$w_{ij}(n+1) = w_{ij}(n) + \Delta w_{ij}(n) \qquad (3.18)$$

where, $\Delta w_{kl}(n), \Delta w_{jk}(n)$ and $\Delta w_{ij}(n)$ are the change in weights of the output, hidden and input layer respectively. That is,

$$\Delta w_{kn}(n) = -2\mu \frac{d\xi(n)}{dw_{kl}(n)} = 2\mu e(n) \frac{dy_l(n)}{dw_{kl}(n)} = 2\mu e(n)\varphi_l'[\sum_{k=1}^{P_2} w_{kl} f_k + \alpha_l] f_k \qquad (3.19)$$

Where, μ is the convergence coefficient $(0 \le \mu \le 1)$. Similarly the can be computed. The thresholds of each layer can be updated in a similar manner, that is

$$\alpha_l(n+1) = \alpha_l(n) + \Delta\alpha_l(n) \qquad (3.20)$$

$$\alpha_k(n+1) = \alpha_k(n) + \Delta\alpha_k(n) \qquad (3.21)$$

$$\alpha_j(n+1) = \alpha_j(n) + \Delta\alpha_j(n) \qquad (3.22)$$

where, $\Delta\alpha_l(n), \Delta\alpha_k(n) and \Delta\alpha_j(n)$ are the changes in thresholds of the output, hidden and input layer respectively. The change in threshold is represented as,

$$\Delta\alpha_l(n) = -2\mu \frac{d\xi(n)}{d\alpha_l(n)} = 2\mu e(n) \frac{dy_l(n)}{d\alpha_l(n)} = 2\mu e(n)\varphi_l'[\sum_{k=1}^{P_2} w_{kl} f_k + \alpha_l] \qquad (3.33)$$

3.3 Radial basis functions networks

The radial basis functions (RBF) network was originally developed for interpolation in multidimensional spaces [7]. Consider a set of m- vectors $\{\rho_i\}$ and a set of associated scalars $\{u_i\}$.The aim is to find a mapping $f : \Re^m \to \Re$ that satisfies

$$u_i = f(\rho_i) \quad , \forall i \qquad (3.34)$$

The function $f(\rho)$ can then be used to interpolate the space $\Re^m$ in all points $\rho \in \Re^m$.The schematic of a RBF network with m inputs and a scalar output is presented in fig. 4.7.

This network can implement a mapping $f_{RBF} : \Re^m \to \Re$, where $f_{RBF}(\cdot)$ is defined as

$$f_{RBF}(z) = \sum_{i=0}^{N_r - 1} w_i \varphi(\|z - \rho_i\|) \qquad (3.35)$$

Where $z \in \Re^m$ is the input vector, $\varphi(\cdot)$ is the *basis* function $\varphi : \Re^+ \to \Re, \rho_i \in \Re^m$ are known as RBF *centres* and w_i are the weights of the centers. That is, the distance of the input z to the corresponding center ρ_i is first extracted and then a scalar function $\varphi(\cdot)$ is calculated onto this distance. This operation is radially symmetric for an Euclidean vector norm $||\text{-}||$, which gives rise to the name of this network.

Some common choices for the basis function $\varphi(\zeta)$ include

a thin plate spline

$$\varphi(\zeta) = \frac{\zeta}{\sigma_r^{\,2}} \log\left(\frac{\zeta}{\sigma_r}\right) \qquad (3.36)$$

a multi quadratic,

$$\varphi(\zeta) = \sqrt{(\zeta^2 + \sigma_r^{\,2})} \qquad (3.37)$$

an inverse multi–quadratic,

$$\varphi(\zeta) = \frac{1}{\sqrt{(\zeta^2 + \sigma_r^{\,2})}} \qquad (3.38)$$

and Gaussian kernel,

$$\varphi(\zeta) = \exp\left(-\frac{\zeta^2}{2\sigma_r^2}\right) \qquad (3.39)$$

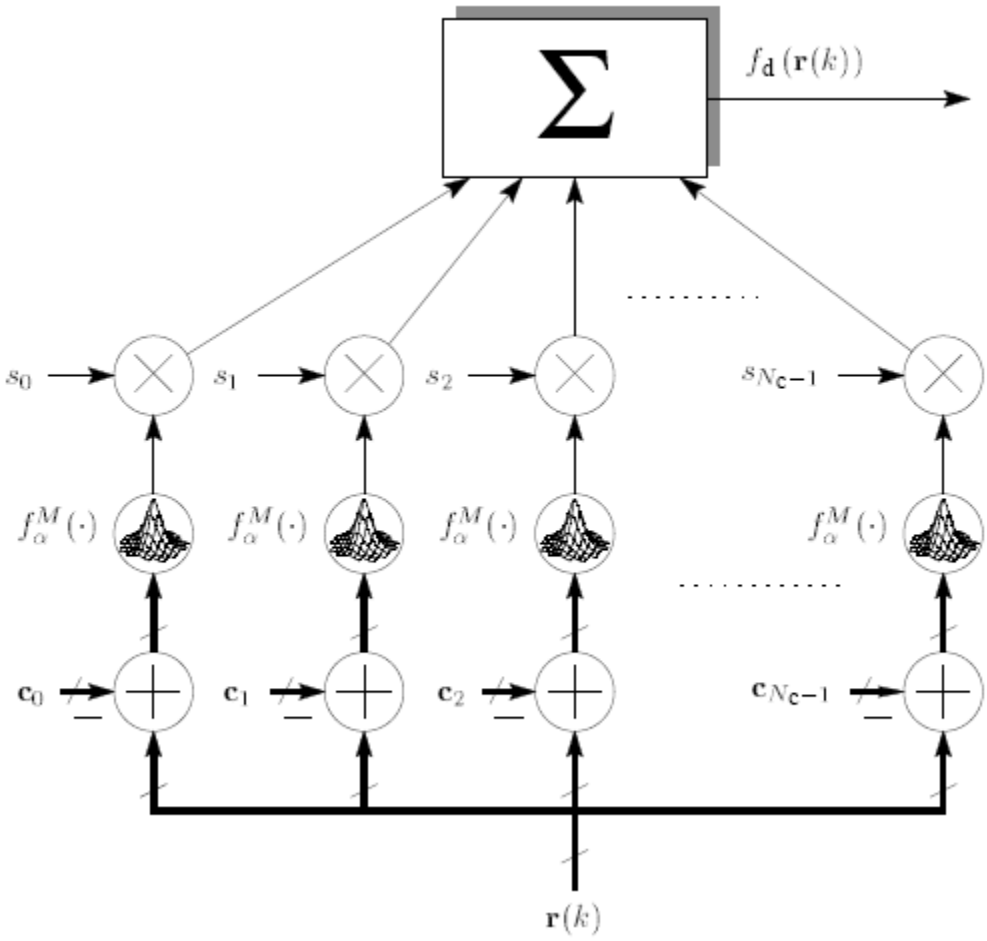


Fig.3.5 A radial basis function network for multidimensional interpolation

The parameter σ_r^2 controls the *radius* of influence of each basis function. The Gaussian and the inverse multi–quadratic kernel, in particular, are bounded and localized, in the sense that the basis functions decay to zero as $\zeta \to \infty$.

Broomhead and Lowe reinterpreted the RBF network as a least square estimator which led to its wide use in signal processing applications such as time series prediction , system identification, interference cancellation, radar signal processing, pattern classification and channel equalization . Training of the RBF networks involves the parameters for the centres ρ_i , radius σ_r and the linear weights w_i .The RBF networks are far easier to train compared to multilayer neural networks, since the training of centre's, radius parameter and the weights can be done sequentially. The main characteristic of the RBF network is that it offers a nonlinear mapping, maintaining at the same time its linearity in parameter structure at the output layer.

3.4 COMPARISON OF RBF NETWORKS AND MLP

Radial Basis Function (RBF) networks and multilayer perceptrons (MLP) are examples of non-linear layered feed forward networks. They are both universal approximators. However, these two networks differ from each other in several important respects.

1) A Radial Basis Function (RBF) (in its most basic form) has a single hidden layer, whereas an MLP may have one or more hidden layers.

2) Typically the computation nodes of an MLP, located in a hidden or an output layer, share a common neuronal model. On the other hand, the computation nodes in the hidden layer of an Radial Basis Function (RBF) network are quite different and serve a different purpose from those in the output layer of the network.

3) The hidden layer of a Radial Basis Function (RBF) network is non-linear, whereas the output layer is non-linear. However, the hidden and output layers of

an MLP used as a pattern classifier are usually all nonlinear .When the MLP is used to solve nonlinear regression problems, a linear layer for the output is usually the preferred choice.

4)The argument of the activation function of each hidden unit in an Radial Basis Function(RBF) network computes the Euclidean norm(distance)between the input vector and the centre of that unit. Meanwhile, the activation function of each hidden unit in an MLP computes the inner product of the input vector and the synaptic weight vector of that unit.

5) The Radial Basis Function (RBF) network is a local approximator, whereas the MLP is a global approximator. i.e., MLPs construct global approximations to nonlinear input-output mapping. On the other hand, Radial Basis Function (RBF) networks using exponentially decaying localized nonlinearities (e.g., Gaussian functions) construct local approximations to nonlinear input-output mappings.

3.5 SUMMARY

In this chapter, a detailed study of the standard form of least-mean-square (LMS) algorithm, back propagation algorithm and Radial Basis Function (RBF) network is being presented. It was found that just as the LMS algorithm has established itself as the workhorse of linear adaptive filters, so it is with the back-propagation algorithm in the context of neural networks. The back-propagation algorithm is relatively simple to implement, which has made it the most popular algorithm in use today for the design of neural networks. But both least-mean-square (LMS) algorithm and back propagation algorithm is characterized by a slow rate of convergence to a local or global minimum of the error performance surface. This limitation is a direct consequence of the fact that both the algorithm operates entirely on the basis of first order information, namely, the gradients of the error-performance surface with respect to the adjustable parameters (weights).

3.5 SUMMARY

The structure of a Radial Basis Function (RBF) network is unusual in that the constitution of its hidden units is entirely different from that of its output units. Unlike multilayer perceptions trained with the back-propagation algorithm, the design of Radial Basis Function (RBF) networks follows a principled approach. Radial Basis Function (RBF) converges faster in comparison to MLP but the computational complexity increases.

CHAPTER 4.

ADAPTIVE CHANNEL EQUALIZATION BY USING LMS ALGORITHM

4. ADAPTIVE CHANNEL EQUALIZATION BY USING LMS ALGORITHM

4.1 INTRODUCTION

In this project the use of LMS algorithm for adaptive equalization of a linear dispersive channel that produces (unknown) distortion is studied. The LMS algorithm is a linear adaptive filtering algorithm which consists of two basic processes:-

a) A filtering process, which involves (1) computing the output of a linear filter in response to an input signal and (2) generating an estimation error by comparing this output with a desired response.

b) An adaptive process, which involves the automatic adjustment of the parameters of the filter in accordance with the estimation error.

4.2 SYSTEM MODEL:

The LMS algorithm for adaptive equalization of a linear dispersive channel is considered .The block diagram of the adaptive equalizer system is shown in Fig.4.1.

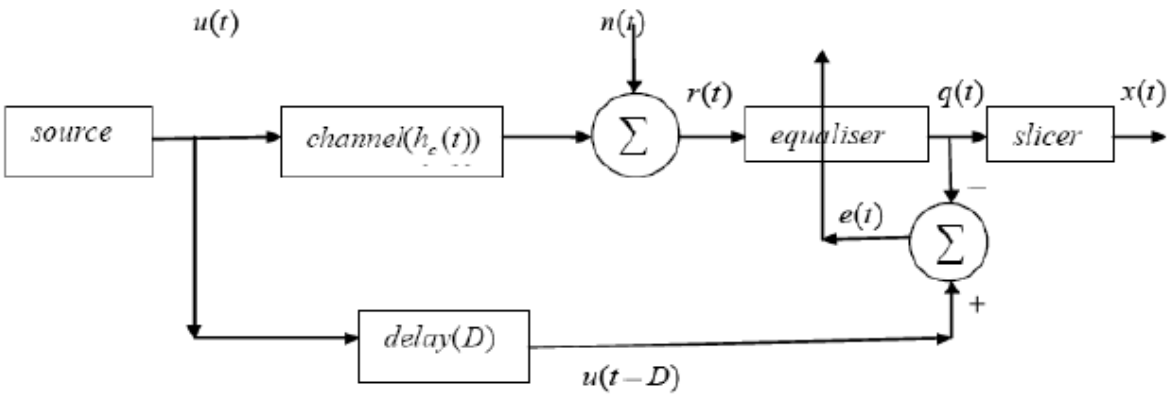


Fig.4.1 Block diagram of adaptive equalizer system

The *source block* transmits Binary Phase Shift Keying (BPSK) symbols *u (t)* = ±1 with equal probability. n (t) = AWGN that corrupts the channel output and has zero mean and variance σ_n^2 =0.001. u (t) and n (t) are independent of each other. The adaptive equalizer corrects the distortion produced by the channel in the presence of the AWGN. The transmitted symbol after suitable delay acts as desired response & is applied to the adaptive equalizer in the form of a training sequence.

The channel model is given in Fig.4.2.

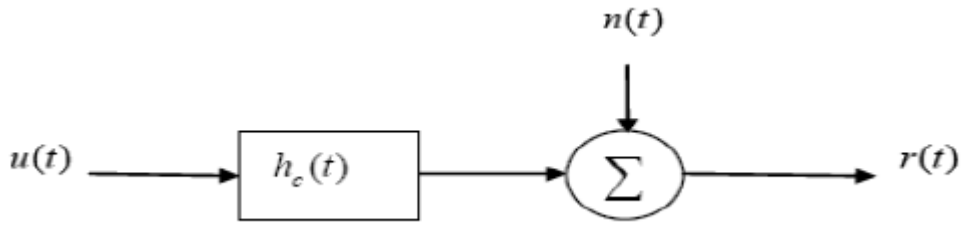


Fig.4.2 Block diagram of Channel model

where, the impulse response of the channel is described by the raised cosine.[1]

$$h_c(t) = \begin{cases} 0.5(1+\cos[2\pi(t-2)/W, t=1,2,3 \\ \\ 0 \text{ , otherwise} \end{cases} \qquad (4.1)$$

Where, the parameter W controls the amount of amplitude distortion produced by the channel.

4.2.1 FIR model of the channel

An ideal physical propagation channel should behave like an ideal low pass filter with fixed amplitude and linear phase characteristics. But, in reality all physical channels deviate from this behavior. When signals are transmitted over a channel, both distortion and additive noise are introduced into it. The transmitted symbols persist beyond the time interval allocated for the transmission and subsequent symbols interfere, causing Inter Symbol Interference(ISI).It is common to model a propagation channel by a digital finite impulse (FIR) filter shown in Fig.(4.3) ,with taps chosen at the signal's sampling interval and coefficients chosen to accurately model the channel impulse response.[1]

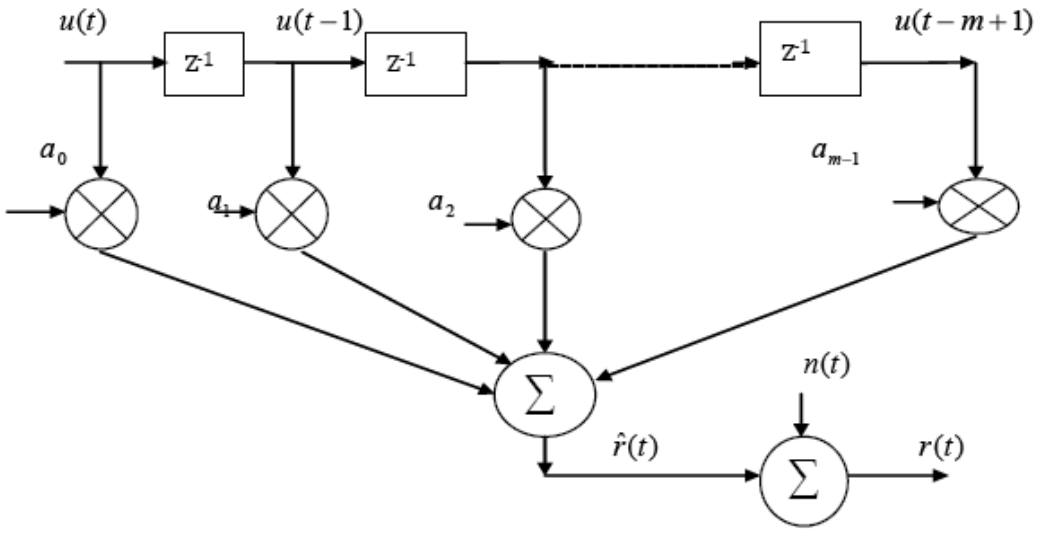


Fig.4.3 FIR model of the Channel

The channel impulse response in z domain can be represented by

$$H(z) = \sum_{i=0}^{m-1} a_i z^{-i} = a_0 + a_1 z^{-1} + a_2 z^{-2} + \ldots\ldots \qquad (4.2)$$

Where m represents the length of the channel impulse response (channel order) and the channel provides dispersion up to m samples. The output from FIR modeled channel is described as

$$r(t) = \hat{r}(t) + n(t) \qquad (4.3)$$

where, r (t) is the channel observed output (input to the equalizer).It is given by the sum of the noise free channel output $\hat{r}(t)$ and AWGN n(t).

For ith symbol transmitted at bit interval T equation (3) can be rewritten as

$$r(t) = \sum_{i=0}^{m-1} a_i u(t - iT) + n(t) \qquad (4.4)$$

$$r(t) = \sum_{i=0}^{m-1} a_i u(t - iT) + n(t) \qquad (4.5)$$

where $\hat{r}(t)$ is the convolution of the transmitted sequence u(t) with the channel taps

a_i, $0 \leq i \leq m-1$.

If u (t) is considered as a rectangular pulse $c_T(t)$ and as binary signaling is employed then r^(t)represents the convolution of the rectangular pulse with the1st order low pass filter impulse response $h_c(t)$.

$\hat{r}(t)$ represents the smeared pulse c (t).The output signal is sampled periodically at sampling time dt to produce the sequence

$$r_l = \sum_{i=0}^{m-1} a_i u_{l-i} + n_l \qquad (4.6)$$

$$= u_0 a_i + \sum_{i \neq l} a_i u_{l-i} + n_l \qquad (4.7)$$

Where, $u_i = u(it), i = 0, \pm 1, \pm 2....$

The second term on the RHS of equation (4.7) represents ISI. Equation (4.7) forms the input to the equalizer.

4.2.2 EQUALIZER MODEL:

An adaptive equalizer using the mean square error (MSE) criterion is designed. The minimization of the cost function can be performed adaptively by applying the stochastic gradient (SG) or the least mean square (LMS) algorithm. The block diagram of a linear adaptive transversal filter is shown in Fig. (4.4).

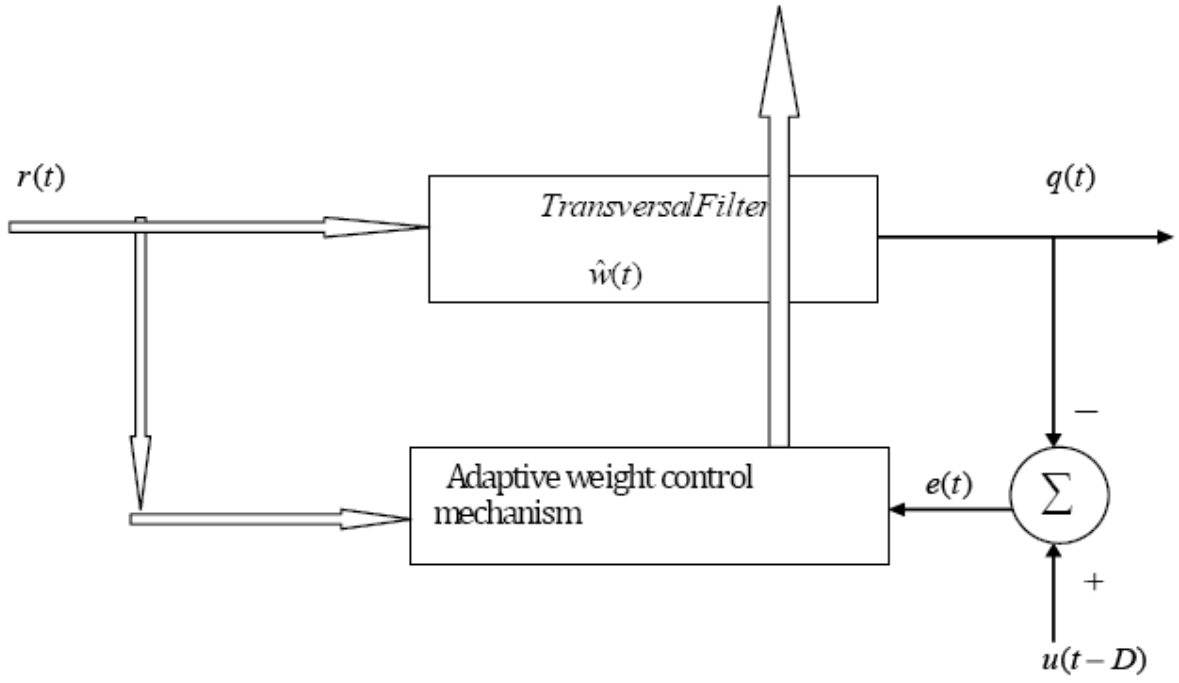


Fig.4.4 Block diagram of Adaptive Transversal Filter

In Fig. (4.4) we have a transversal filter, around which the LMS algorithm is built; this component is responsible for performing the filtering process. The adaptive weight control mechanism is responsible for performing the adaptive control process on the tap weights of the transversal filter.

The detailed structure of the transversal filter and the adaptive weight control mechanism is shown in Fig. (4.5).

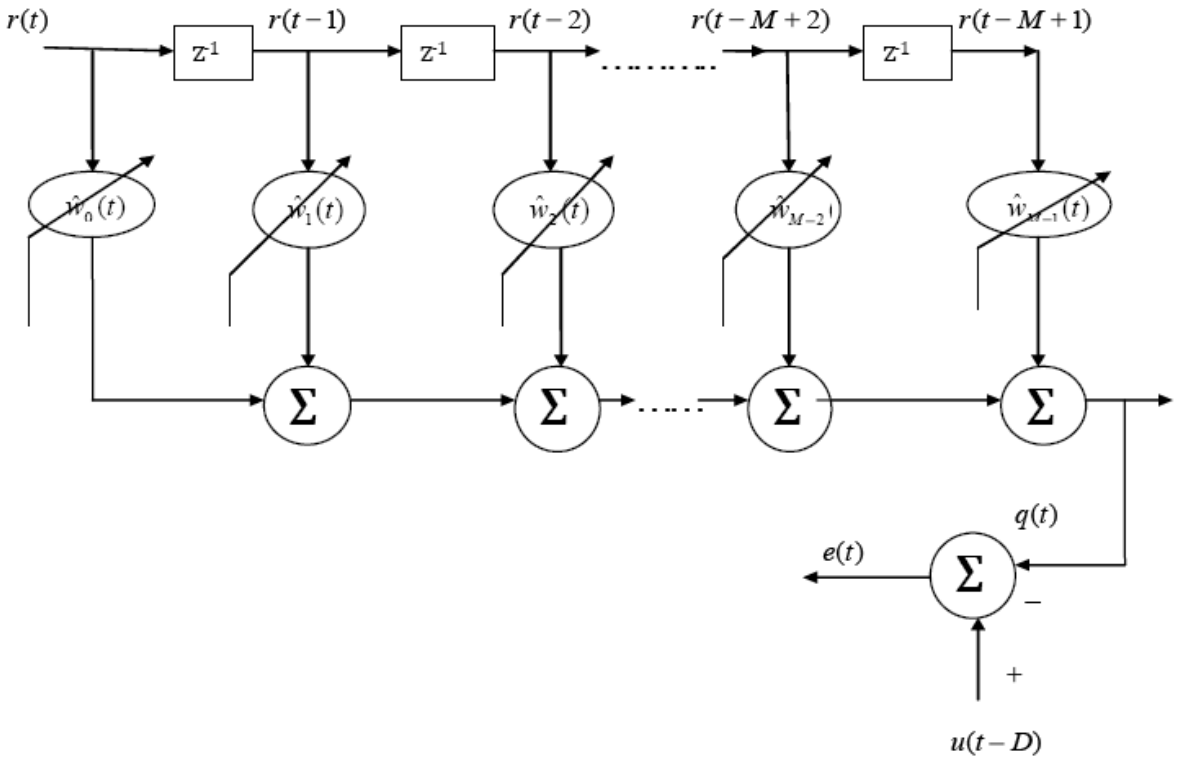


Fig.4.5 Detailed Structure of the Transversal Filter

The tap inputs r (t), r (t-1),….,r(t-M+1) form the elements of the M-by-1 tap-input vector r(t),where M-1 is the number of delay elements. Correspondingly, the tap weights $\hat{w}_0(t)$, $\hat{w}_1(t)$,……. $\hat{w}_{M-1}(t)$ form the elements of

the M-by-1 tap-weight vector $\hat{w}(t)$.During the filtering process, the desired response u(t-D) is supplied for processing, alongside the tap-input vector r(t).

Given this input, the transversal filter produces an output q(t) used as an estimate of the desired response u(t-D).Hence estimation error e(t) is defined as the difference between the desired response and the actual filter output as indicated in Fig.(4.5) .

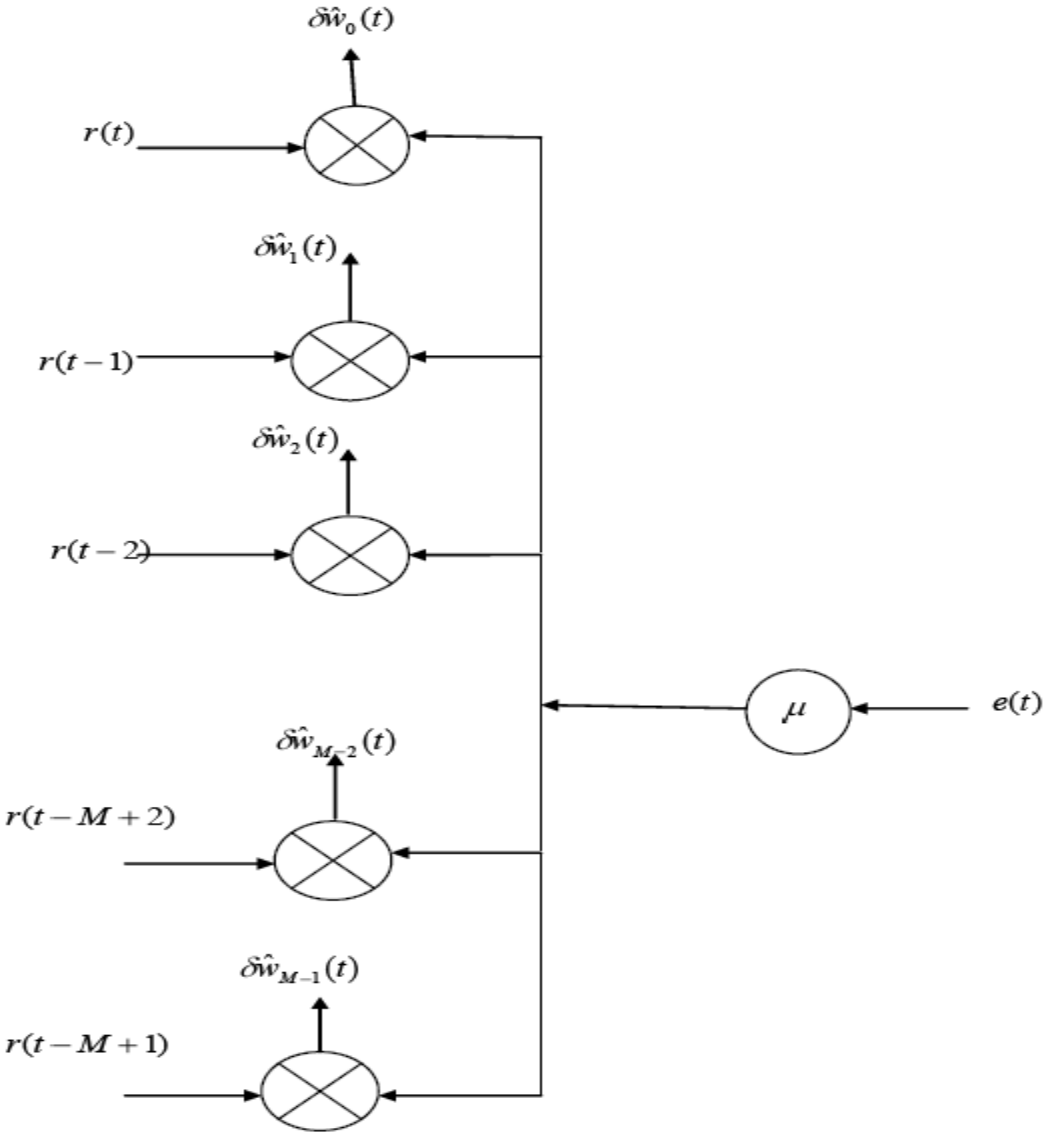


Fig.4.6 Detailed Structure of the Adaptive weight- control mechanism

A scalar version of the inner product of the estimation error e(t) and the tap input r(t-k) is computed for k=0,1,2,…..M-2,M-1.The result so obtained defines the correction $\delta\hat{w}_k(t)$ applied to the tap weight $\hat{w}_k(t)$ at iteration t+1.The scaling factor used in this computation is denoted by a positive quantity μ in figure(4.6) called the step-size parameter.

4.3. LMS ADAPTATION ALGORITHM

It can be expressed in the form of three basic relations as follows:

1. Filter output:

$$q(t) = \hat{w}^H(t)r(t) \qquad (4.8)$$

2. Estimation error or error signal:

$$e(t) = u(t-D) - q(t) \qquad (4.9)$$

3. Tap-weight adaptation:

$$\hat{w}(t+1) = \hat{w}(t) + \mu r(t)e(t) \qquad (4.10)$$

Equations(4.8) and (4.9) defines the estimation error e(t),the computation of which is based on the current estimate of the tap-weight vector , $\hat{w}(t)$ .The second term ,on the RHS of equation(4.10) represents the adjustment that is applied to the current estimate of the tap-weight vector, $\hat{w}(t)$ .The iterative procedure is started with an initial guess $\hat{w}(0)$.

4.4 SIGNAL FLOW GRAPH OF THE LMS ALGORITHM

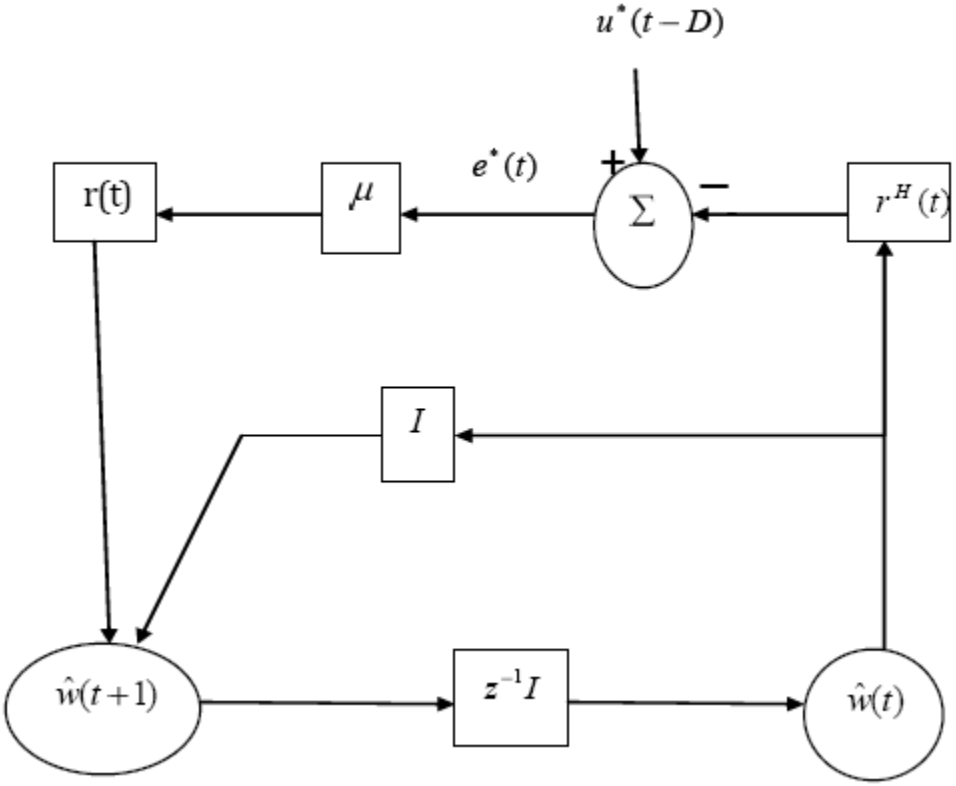


Fig.4.7 Signal-flow graph representation of the LMS algorithm.

4.5 SIMULATION PARAMETERS

The equalizer has M = 11taps.As the channel has an impulse response $h_c(t)$ that is symmetric about time t = 2 & the optimum tap weight s w_{on} of the equalizer are likewise symmetric about time t = 5.The channel input u(t) is delayed by D = 2 + 5 = 7 samples .

to provide the desired response for the equalizer [1].By selecting the delay Δ to match the midpoint of the transversal equalizer, the LMS algorithm is enabled to provide an approximate inversion of both the minimum- phase and non-minimum- phase components of the channel response.

The simulation is carried out to evaluate the response of the adaptive equalizer using the LMS algorithm to changes in the step-size parameter μ.

4.5.1 Correlation matrix of the equalizer input.

The first tap input of the equalizer at time t is

$$r(t) = \sum_{v=1}^{3} h_v u(t-v) + n(t) \qquad (4.11)$$

where all the parameters are real valued. Hence, the correlation matrix R of the 11 tap inputs of the equalizer, r(t), r(t – 1).....r(t – 10) ,is a symmetric 11-by-11 matrix. Also, since the impulse response $h_c(t)$ has non-zero values only for t = 1,2,3 & the noise process n(t) is white with zero mean & variance σ_n^2. the correlation matrix R is quintdiagonal,i.e., the only non-zero elements of R are on the main diagonal and t he four diagonals directly above & below it, two on either side as shown below:

$$R = \begin{vmatrix} r(0) & r(1) & r(2) & 0 \ldots 0 \\ r(1) & r(0) & r(1) & r(2) \ldots 0 \\ r(2) & r(1) & r(0) & r(1) \ldots 0 \\ 0 & r(2) & r(1) & r(0) \ldots 0 \\ \vdots & & & \\ 0 & 0 & 0 & 0 \ldots r(0) \end{vmatrix}$$

Where, $r(0) = h_1^{\,2} + h_2^{\,2} + h_3^{\,2} + \sigma_n^{\,2}$

$r(1) = h_1 h_2 + h_2 h_3$

and $r(2) = h_1 h_3$

The variance $\sigma_n^{\,2} = 0.001$; hence, h_1, h_2 and h_3 are determined by the value assigned to the parameter W in equation (4.1).

4.5.2 Effect of step-size parameter(μ)

For W=2.9, we have the following values for the autocorrelation function r (l) for lag l=0,1,2 and the smallest eigen value $\lambda_{\min}$ and the largest eigen value $\lambda_{\max}$ and the eigen value spread χ (R).

r (0)=1.0963,r (1)=0.4388,r (2)=0.0481, $\lambda_{\min}$=0.3339, $\lambda_{\max}$=2.0295 and

χ (R)= $\lambda_{\max}/\lambda_{\min}$=6.0782.

No. of independent runs I=200

No. of transmitted symbols K=3000

No. of filter coefficients M=11

Noise power= $\sigma_n^{\,2} = 0.001$

Channel coefficients= [h_0, h_1, h_2]=[0.2194 1.0000 0.2194]

The step-size parameter was assigned one of the three values 0.025, 0.01, 0.0075.

The mean square error (MSE) for the filter in the t^{th} time instant is defined as

$MSE=E[|e|^2]$ ------(4.12)

In simulations,the expectation in (4.12) is evaluated through averaging over I independent runs,in order to be able to view the convergence of the equaliser as a function of time.The equaliser tap weights are initialised to zero.

4.6 SIMULATION RESULTS

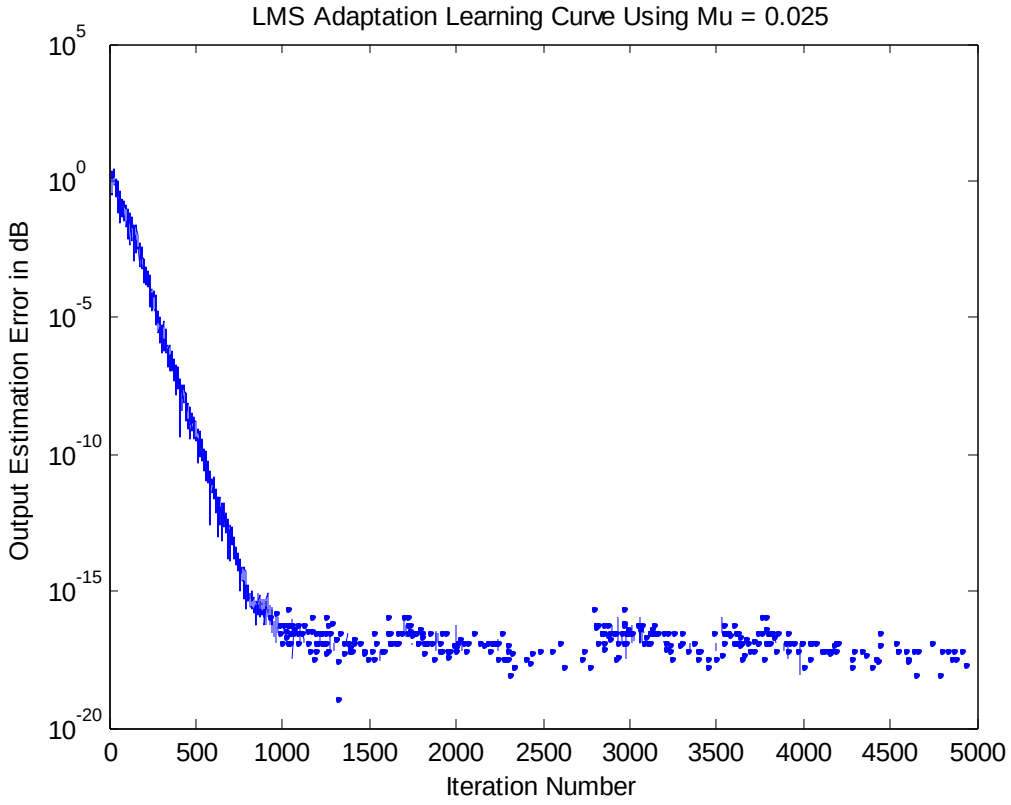


Fig. 4.8 LMS Adaptation Learning Curve with Step-Size Parameter=0.025

The learning curve converged to steady state condition in approximately 1000 iterations and the mode of convergence is hardly visible.

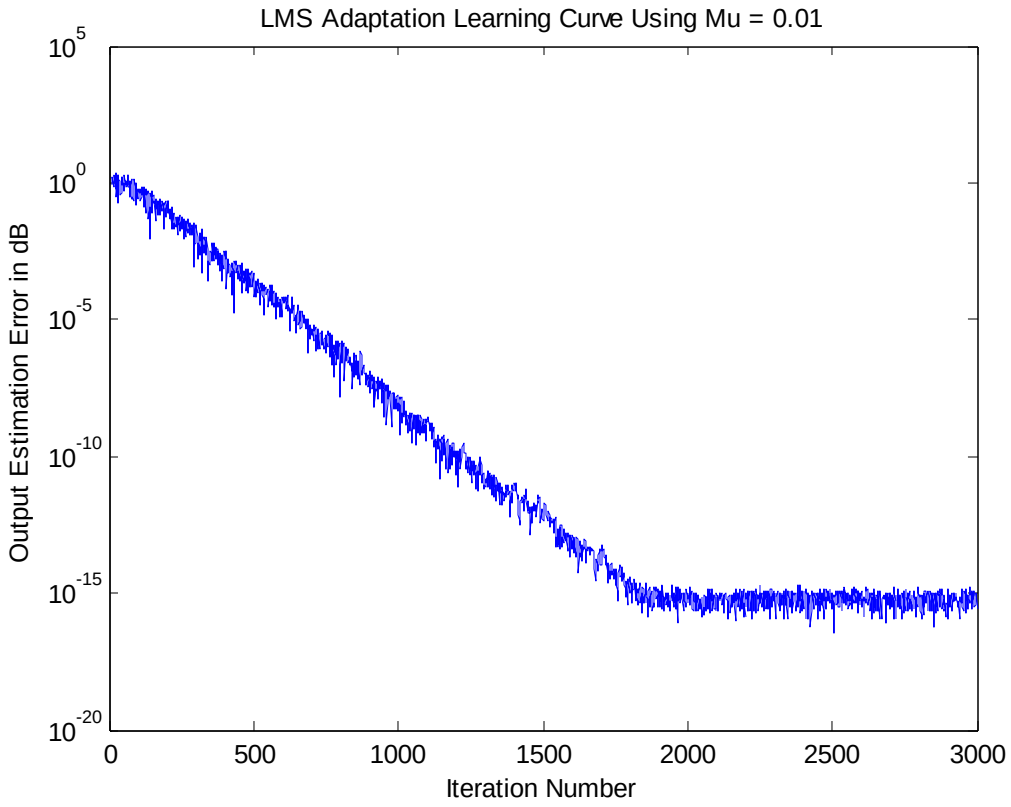


Fig. 4.9 LMS Adaptation Learning Curve with Step-Size Parameter=0.01

The learning curve converged to steady state condition in approximately 1800 iterations.

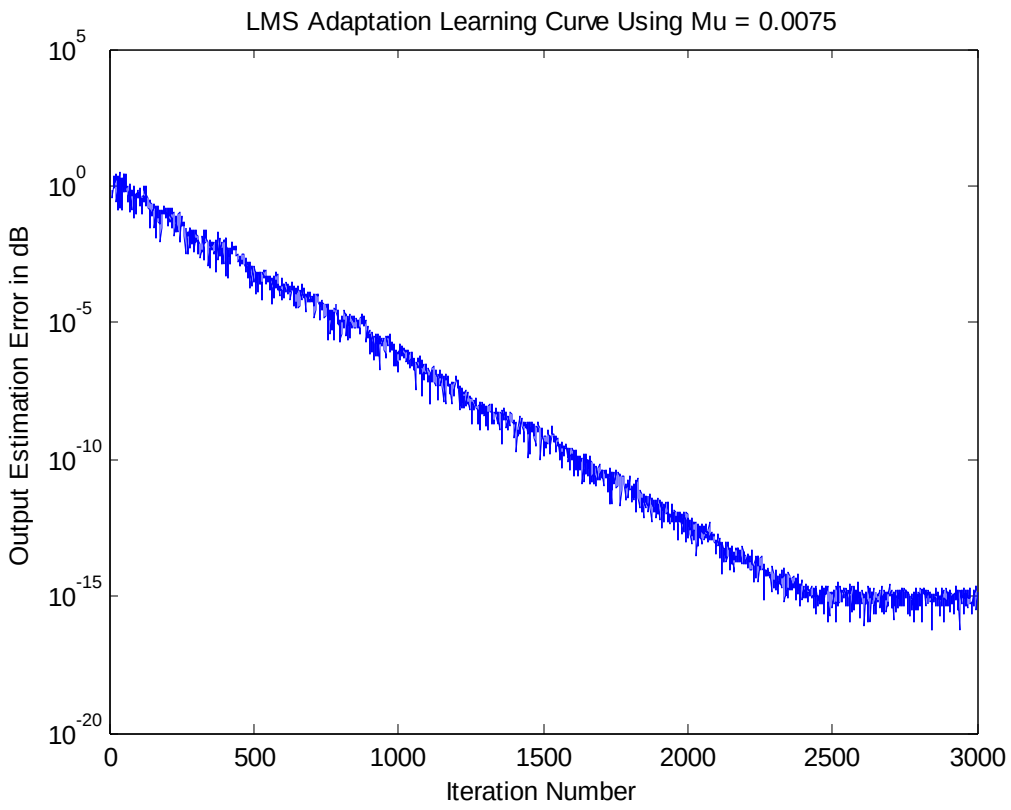


Fig. 4.10 LMS Adaptation Learning Curve with Step-Size Parameter=0.0075

The learning curve converged to steady state condition in approximately

2500 iterations.

4.7. DISCUSSION

Each learning curve is the result of ensemble averaging the instantaneous squared error "e^2 (t) versus t" curve over 200 independent trials. The results confirm that the rate of convergence of the adaptive equalizer is highly dependent on the step-size parameter μ.For a large step-size parameter (μ=0.025), the equalizer converged to steady-state conditions in approximately 1000 iterations. On the other hand, when μ was small (=0.0075), the rate of convergence slowed down to 2500 iterations. The result also shows that the learning curve of the channel equalizer is pre-dominantly controlled by its slower

modes of convergence (when μ=0.0075) while the fast mode of convergence is hardly visible (when μ=0.025).It has been observed that the convergence of the faster modes of the LMS algorithm has not reduced MSE significantly. It is due to the fact that the filter length M is small.

4.8 Summary

In this chapter, the least mean square filters (LMS), which are the workhorses of linear adaptive filtering has been studied. The practical implementation of LMS filters is due to the following facts:

1. Simplicity of implementation

2. A model-independent and therefore robust performance.

The main limitation of LMS filters is their relatively slow rate of convergence. One of the principal factors that affect the convergence behavior of an LMS filter is the step-size parameter μ.

When a small value is assigned to μ, the adaptation is slow .On the other hand, when a large value is assigned to μ, the adaptation is relatively fast.

Chapter 5.

Decision Feedback Equalization Using LMS and MLP

5. Decision Feedback Equalization Using LMS and MLP

5.1. Introduction

Decision feedback equalization is a technique used in digital communications systems (Fig.5.1) to equalize the channel to remove that part of the intersymbol interference (ISI) caused by the previous data decisions.

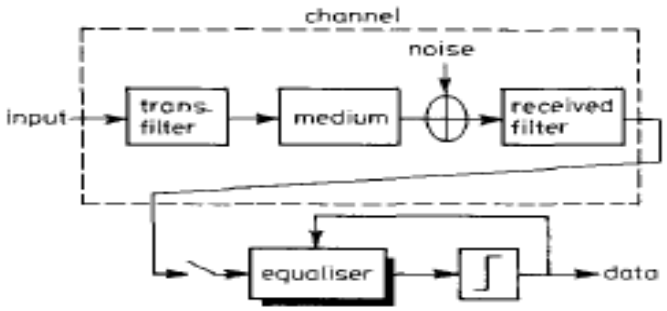


Fig.5.1 Baseband data transmission system

The advantage of the decision feedback equalizer is that IS1 is eliminated without enhancement of noise by using past decisions to subtract out a portion of the IS1 in addition to the normal feed forward filter; a disadvantage is that decision errors tend to propagate because they result in residual IS1 and a reduced margin against noise at future decisions [l]. The conventional structure of the decision feedback equalizer (DFE) uses linear algorithms such as LMS (least mean square) or RLS (recursive least square) and consists of a feed forward filter and a feedback filter, as shown in Fig.5.2, where the feed forward filter is a linear, equalizer. The decision regions of a linear equalizer are always delimited by hyper planes. The linearity of these decision boundaries limit the performance of the system.

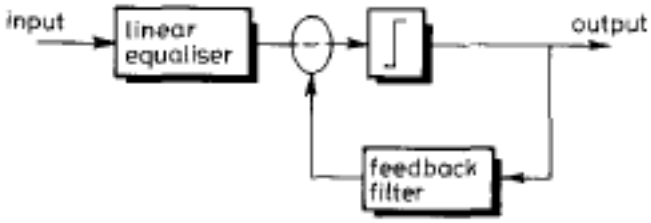


Fig.5.2 Decision feedback equalizer structure

Artificial neural networks [17] are systems which use nonlinear computational elements to model neural behavior based on our present understanding of the biological nervous system. A neural network may be simply considered as a nonlinear mapping between input and output. Rumelhart et al. (1986) [18] proposed a back propagation learning algorithm enabling multilayer perception networks [18], [19], [20] to learn more sophisticated tasks than before. The network uses a layered feed forward structure with input, output and hidden layer(s). The hidden layers provide the capability by use of the nonlinear sigmoid function, to create intricately curved partitioning of the signal space to produce nonlinear decision boundaries [2, 22].

5.2. Multilayer perceptrons: Architecture

The basic element of the multilayer perceptron is the neuron, which is depicted in Fig. 5.3. Each neuron has primarily local connections and is characterized by a set of

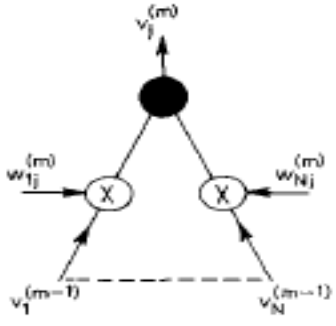


Fig.5.3 jth neuron in m^{th} layer

real weights $[w_{1j},.....,w_{Nj}]$ applied to the previous layer to which it is connected and a real threshold level I_j. The j[th] neuron in the m[th] layer accepts inputs $V^{(m-1)} \in R^N$ from the (m - 1)[th] layer and returns a scalar $v_j^{(m)} \in R$ given by

$$v_j^{(m)} = f_j\left(\sum_{i=1}^{N} w_{ij}^{(m)} v_i^{(m-1)} + I_j^{(m)}\right) \qquad (5.1)$$

The output value $v_j^{(m)}$ serves as input to the (m + 1)[th] layer to which the neuron is connected.

The nonlinearity commonly used in the perceptron is of the sigmoid type:

$$f(x) = \frac{1-e^{-x}}{1+e^{-x}} \qquad (5.2)$$

Where $f(x)$ lies in the interval [- 1, 1] as shown in Fig.5.4.

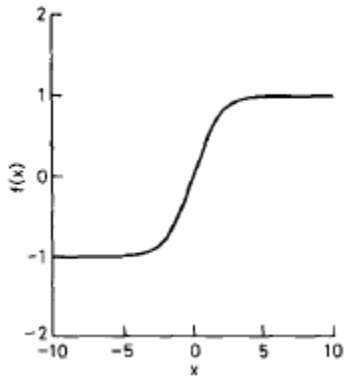


Fig.5.4 Activation function

The neurons store knowledge or information in the weights $\{w_{ij}\}$ and the weights are modified through experience or training.

Earlier work on single layer perceptrons was limited owing to the fact that only linear decision boundaries could be formed in the signal space [17]. However, later developments [18-20, 24] showed how multiple layers could be used to form much more complex (nonlinear) decision boundaries. A multilayer perceptron (MLP) consists of several hidden layers of neurons which are capable of performing complex, nonlinear mappings between the input and the output layer. The hidden layers provide the capability to use the nonlinear sigmoid's ability to create intricately curved partitions of space. In general, all neurons in a layer are fully interconnected to neurons in adjacent layers, but there is no connection within a layer, and normally no connections bridging layers, as shown in Fig.5.5. Data information is recoded into the hidden layer(s) and the output is generated by combinational operations on the final hidden layer.

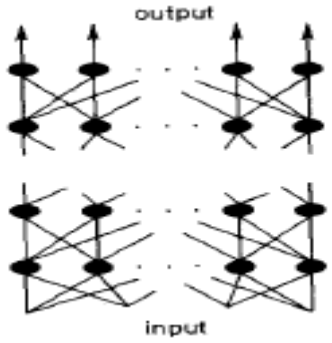


Fig.5.5 Multilayer perceptron architecture

5.3 Perceptron-based decision feedback equalizer

A three-layer preceptron based decision feedback equalizer structure, as shown in Fig. 5.6, consists of a feed- forward filter and a feedback filter. The input to the feed forward filter is the sequence of noisy received signal samples $\{y_n\}$. The input to the feedback filter is the output symbol decision sequence from a nonlinear symbol detector (quantizer) $\{\tilde{u}_{n-d}\}$.

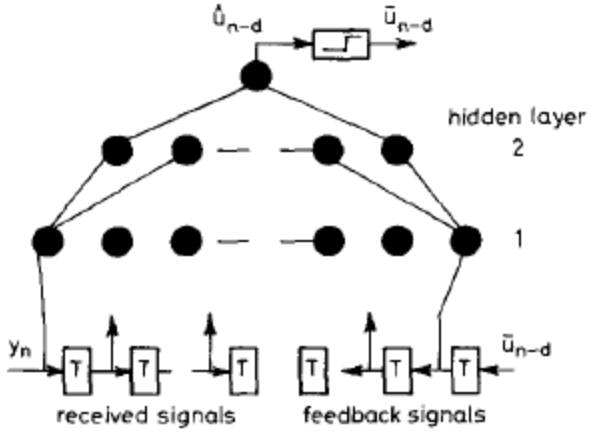


Fig.5.6 Multilayer perceptron decision feedback equalizer

The three-layer perceptron (two hidden layers, and an output layer) is sufficient for the nonlinear DFE structure, because a three-layer perceptron can generate arbitrarily complex, nonlinear decision regions [22].

At time *n,* the input *N* x 1 received signal vector

$$\Gamma(n)^T = [y_n, y_{n-1}, \ldots\ldots, y_{n-N+1}] \qquad (5.3)$$

and the decision $l \times 1$ signal vector

$$[\tilde{u}_{n-d-1}, \tilde{u}_{n-d-2}, \ldots\ldots, \tilde{u}_{n-d-l}] \qquad (5.4)$$

are in the feed forward filter and feedback filter of the decision feedback equalizer, respectively, where d is a delay parameter. The decision $\tilde{u}_{n-d}$ is formed by quantizing the estimate $\tilde{u}_{n-d}$ in the output layer to the nearest information symbol.

The signals at the input layer of the decision feedback equalizer can be represented by a $(N+l)\times 1$ vector as

$$V^{(0)} = [y_{n,}y_{n-1},\ldots,y_{n-N+1};\tilde{u}_{n-d-1},\ldots.\tilde{u}_{n-d-l}]^T \quad (5.5)$$

The $N_1 \times 1$ vector in the output of hidden layer 1 is

$$V^{(1)} = [v_1^{(1)}, v_2^{(1)}, \ldots\ldots\ldots, v_j^{(1)}, \ldots\ldots\ldots, v_{N_1}^{(1)}]^T \quad (5.6)$$

Where

$$v_j^{(1)} = f_j\left(\sum_{i=0}^{N-1} w_{ij}^{(1)} y_{n-i} + \sum_{p=1}^{l} w_{pj}^{b(1)} \tilde{u}_{n-d-p} + I_j^{(1)}\right) \quad j = 1,2,\ldots\ldots N_1 \quad (5.7)$$

Where b denotes the feedback tap weight.

The $N_2 \times 1$ vector in the output of hidden layer 2 is

$$V^{(2)} = [v_1^{(2)}, v_2^{(2)}, \ldots. v_k^{(2)}, \ldots., v_{N_2}^{(2)}]^T \quad (5.8)$$

Where

$$v_k^{(2)} = f_k\left(\sum_{j=1}^{N_1} w_{jk}^{(2)} v_j^{(1)} + I_k^{(2)}\right) \qquad k = 1,2,\ldots\ldots N_2 \quad (5.9)$$

The final output is

$$v_0^{(3)} = \hat{u}_{n-d} = f_0\left(\sum_{k=1}^{N_2} w_{k0}^{(3)} v_k^{(2)} + I_0^{(3)}\right) \quad (5.10)$$

Where $\hat{u}_{n-d}$ is the estimated signal at time *n*. Substituting eqns. 5.7 and 5.9 into eqn. 5.10, yields

$$\hat{u}_{n-d} = f_0\left(\sum_{k=1}^{N_2} w_{k0}^{(3)} f_k\left(\sum_{j=1}^{N_1} w_{jk}^{(2)} f_j\left(\sum_{i=0}^{N-1} w_{ij}^{(1)} y_{n-i} + \sum_{p=1}^{l} w_{pj}^{b(1)} \tilde{u}_{n-d-p} + I_j^{(1)}\right) + I_k^{(2)}\right) + I_0^{(3)}\right) \quad (5.11)$$

The nonlinear detector can be modeled as a threshold function $g(x)$ and is defined as

$$g(\hat{u}_{n-d}) = \tilde{u}_{n-d} = \begin{cases} 1 \quad \text{if} \quad \hat{u}_{n-d} \geq 0 \\ -1 \quad \text{otherwise} \end{cases} \qquad (5.12)$$

The ws (weights) and Is (threshold levels) in eqn. 5.11 are values specified by the training algorithm, so that after training is finished the equalizer will self-adapt to changes in channel characteristics occurring during transmission (decision directed mode).

5.4. Eliminating intersymbol interference: decision feedback signal

The output $v_j^{(1)}$, of the j[th] neuron in layer one can be expressed in terms of $\{g_p\}$,the feedback tap weights $\{w_{pj}^{b(1)}\}$ and the transmitted signal $\{u_n\}(u_n \in (1,-1))$ as shown in Fig.5.7. Note that $\{g_p\}$ is the convolution of the channel impulse response $\{h_p\}$ and the weights $\{w_{ij}^{(1)}\}$.Thus

$$v_j^{(1)} = f_j\left(\sum_p u_{n-p} g_p + \sum_{p=1}^{l} w_{pj}^{b(1)} \tilde{u}_{n-p} + \eta_n + I_j\right) \qquad (5.13)$$

Where η_n is zero-mean Gaussian noise. The above equation can be written as

$$v_j^{(1)} = f_j\left(u_n g_0 + \sum_{p=1}^{l}\left(g_p + w_{pj}^{b(1)}\right)\tilde{u}_{n-p} + \sum_{p=1}^{l}\left(u_{n-p} - \tilde{u}_{n-p}\right)g_p + \sum_{\substack{p<0 \\ p>l}} u_{n-p} g_p + \eta_n + I_j\right) \qquad (5.14)$$

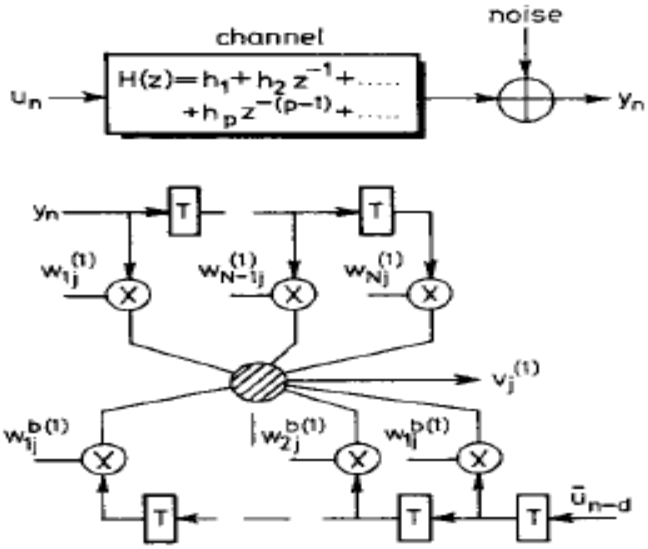


Fig. 5.7 j^{th} neuron with feedback signals in first layer

If we select

$$g_p = -w_{pj}^{\,b(1)} \quad p = 1,2,.....,l \tag{5.15}$$

and the probability of error is very small, we may assume that the last l symbols have been received correctly, i.e. $\tilde{u}_{n-p} = u_{n-p} \quad p = 1,2,.....,l$ (5.16)

Then eqn. (5.14), can be simplified as

$$v_j^{(1)} = f_j\left(u_n g_0 + \sum_{\substack{p<0 \\ p>l}} u_{n-p} g_p + \eta_n + I_j \right) \tag{5.17}$$

All IS1 from past symbols $(1 \le p \le l)$ is eliminated without altering the useful signal term $u_n g_0$ or enhancing the noise component η_n. The $\sum_{\substack{p<0 \\ p>l}} u_n g_p$ residual IS1 term will be reduced as the signal is passed forward. If an incorrect decision is made by the detector, e.g., $\tilde{u}_{n-p} = -u_{n-p}$, the decision errors tend to propagate because they result in residual intersymbol interference and a reduced margin against noise at future decisions.

For an equalizer with (P) taps and a channel response that spans *(L)* symbols, the number of symbols involved in the intersymbol interference is (P + *L* - 2). The number of taps (l) that are needed in the feedback section to eliminate all the ISI from previously detected symbols, provided that previous decisions are correct is

l = P + L - 2 - d (5.18)

where d is a delay parameter.

5.5. Learning algorithm

An iterative learning algorithm, called back propagation was suggested by Rumelhart et al. [3]. In back propagation, the output value is compared with the desired output, resulting in an error signal. The error signal is fed back through the network and weights are adjusted to minimize this error.

The increments used in updating the weights, Δw_{ij} and threshold levels, ΔI_j of the m[th] layer can be accomplished by the following rules:

$$\Delta w_{ij}^{(m)}(n+1) = \eta \delta_j^{(m)} v_j^{(m-1)} + \alpha \Delta w_{ij}^{(m)}(n) \qquad (5.19)$$

and

$$\Delta I_j^{(m)}(n+1) = \beta \delta_j^{(m)} \qquad (5.20)$$

where η is the learning gain, α is the momentum parameter, β is the threshold level adaptation gain, and layer $m \in [1,2,.....,M]$

The error signal $\delta_j^{(m)}$ for layer *m* is calculated starting from the output layer M

$$\delta_j^{(M)} = (t_j - v_j^{(M)})(1 - v_j^{2(M)}) \Big/ 2 \qquad (5.21)$$

and recursively back-propagating the error signal to lower layers

5.6. Perceptron-based DFE performance and comparison with LMS DFE

$$\delta_j^{(m)} = (1 - v_j^{2(m)}) \sum_l \delta_l^{(m+1)} w_{lj}^{(m+1)} \Big/ 2 \qquad (5.22)$$

Where, $m \in [1,2,....M-1]$, l is over all neurons in the layer above neuron j and t_j is the desired output.

To allow rapid learning a momentum term, $\Delta w_{ij}^{(m)}(n)$, scaled by α is used to filter out high frequency variation of the weight vector. As a result, the convergence rate is much faster and the weight changes are smoothed.

5.6. Perceptron-based DFE performance and comparison with LMS DFE

The channel model used in the performance evaluation is given in z-transform notation by $H(z) = 0.3482 + 0.8704\, z^{-1} + 0.3482\, z^{-2}$.

The digital message applied to the channel was in random bipolar form {- 1, 1}. The channel output is corrupted by zero mean white Gaussian noise. For mathematical convenience, we normalize the received signal power to unity. Then the received signal to noise ratio (SNR) is simply the reciprocal of the noise variance at the input of the equalizer.

The performance was determined by taking an average of 600 individual runs. Each run had a different random sequence and random starting weights. For simplicity the short hand notation $\{(N,l)DFE$ with $(N_1, N_2, N_3)MLP\}$ (MLP DFE) will be used to indicate that the number of received signal samples is N, the number of decision feedback samples is l, the number of neurons in hidden layer l(H_l) is N_l, the number of neurons in hidden layer 2(H_2,) is N_2, and the number of neurons in output layer is N_3, for a three layer perceptron based decision feedback equalizer.

5.7 Convergence characteristics

Fig.5.8 illustrates MSE (mean square error) convergence of the MLP DFE, {(4, 1) DFE with (9, 3**,** 1) MLP structure}, with learning gain (η) 0.07 and the LMS DFE with learning gain (μ) 0.035 are shown in Fig.5.8 (a and b). Also the MSE convergence of equalizers for no feedback signals (simple transversal equalizers) are shown in Fig.5.8 (c and d). The MLP DFE requires at least l000 iterations to converge while the LMS DFE converges in about 120 iterations. The results also show that the steady-state value of averaged square error produced by the MLP DFE converges to a value (< -25 dB) which is lower than the additive noise (-20 dB). This is a result of the nonlinear nature of equalizer transfer function. The LMS DFE gives a steady value of averaged square error at about -14.0 dB which is above the noise floor using the same number of input samples. The result also indicates that both types of the decision feedback equalizers yield a significant improvement in convergence time and averaged square error relative to the equalizers without feedback signal having the same number of input samples.

5.8 Decision region

Fig.5.9 shows the decision region formed by a {(2,0)DFE with (9,3,1)MLP structure} (without feedback signal) and the decision boundary formed by the optimal equalizer based on the *maximum* a *posteriori* (MAP) criterion[6], [8].The signal to noise ratio was 10dB. The *Maximum a Posteriori* criterion will yield a minimum probability-of-error decision. It can be seen that the decision region formed by the perceptron is near that optimal decision region which suggests that the perceptron is utilizing the available information with something approaching maximum efficiency.

5.9 Bit error rate performance - decision directed mode

DFE performance can be obtained by means of a Monte Carlo simulation. Fig.5.10 illustrates error rate performance in the stationary channel case for the MLPDFE performance using either correct or detected symbols in the feedback section with η = 0.1. For illustrative purposes the performance of the perceptron based equalizer (without feedback) with η = 0.1 is shown.

It may be observed from Fig.5.10 that the MLP DFE attains about 5 dB improvement at BER =10^{-4} relative to the MLP equalizer having the same number of input samples. The performance loss owing to incorrect decisions being fed back is approximately 1.3 dB for the channel response under consideration. Fig.5.11 illustrates the performance of the LMS DFE with μ = 0.035. The results show that the LMS DFE attains 5 dB improvement at BER =10^{-4} relative to the LMS equalizer. The performance loss owing to incorrect decisions being fed back is about 2.0 dB. From the data in Figs.5.10 and 5.11 it can be seen that the performance degradation owing to decision errors for the perceptron based DFE is less than for the LMS DFE, especially under high noise conditions.

Fig.5.12 illustrates performance for the MLP DFE with η =0.07, 0.1 and the LMS DFE with μ = 0.035, 0.05 having the same no. of input samples. In the simulation, all the symbols fed back are detected symbols. The MLP DFE structure performs the superior performance in comparison with the LMS DFE structure, when the level of additive noise is high, but deteriorates as the signal to noise ratio improves. This latter fact is due to the fact that, if the additive noise level is very low, the MLP DFE will receive very few samples of signal which are close to the optimal decision boundary, rendering it incapable of forming optimal decision boundary as it does in the high noise situation.

5.10 Simulation Results

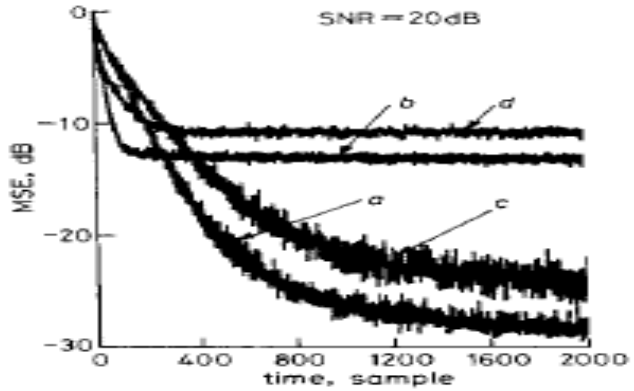


Fig.5.8 Simulation results showing relative Convergence rate performance
(a) (4, 1) DFE with (9, 3, l) MLP structure: η = 0.07, α = 0.3, β = 0.05
(b) LMS (4, 1) DFE structure: μ = 0.035
(c)(5, 0)DFE with (9, 3, l) MLP structure (without feedback signal): η = 0.07, α =0.3, β =0.05
(d) LMS (5, 0) DFE structure (without feedback signal): μ = 0.035 at SNR = 20 dB

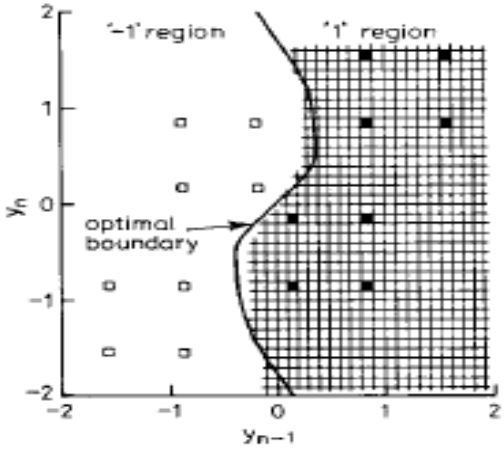


Fig.5.9 Decision region formed by (2, 0) DFE with (9, 3, 1) MLP structure after 300 samples: η= 0.3, α = 0.3 and β = 0.05 where shading denotes decision region for 'l' and optimal decision boundary formed by the MAP criterion with SNR = 10 dB

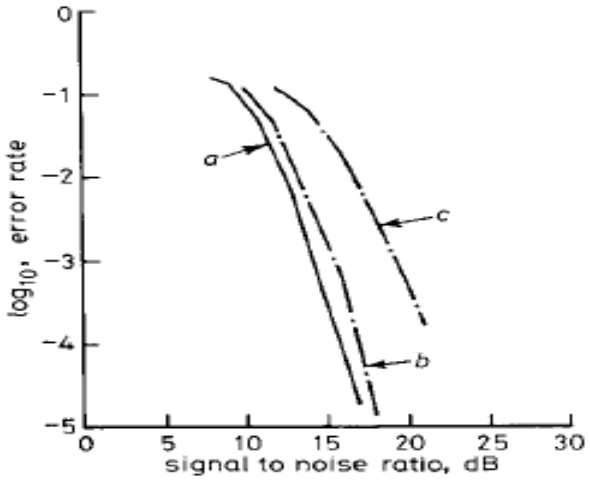


Fig.5.10 Performance of (4, l) DFE with (9, 3, 1) MLP structure with and without error propagation, η = 0.1, α = 0.3, β = 0.05

(a) Correct bit fed back (b) Detected bit fed back

(c) (5, 0) DFE with (9, 3, l) MLP structure (without feedback signal)

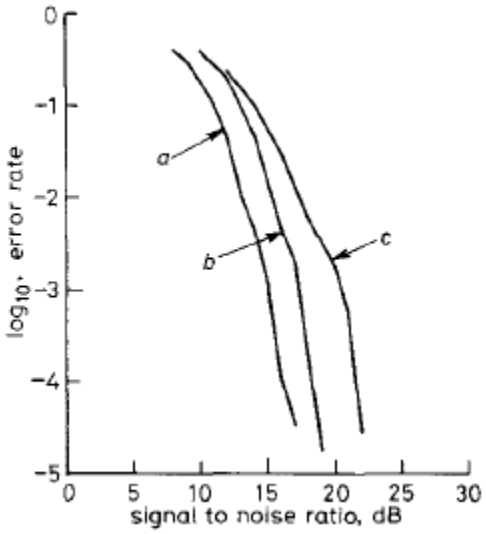


Fig.5.11 Performance of LMS (4, 1) DFE structure with and without error propagation, μ = 0.035

(a) Correct bit fed back

(b) Detected bit fed back

(c) LMS (5, 0) DFE structure (without feedback signal): μ = 0.035

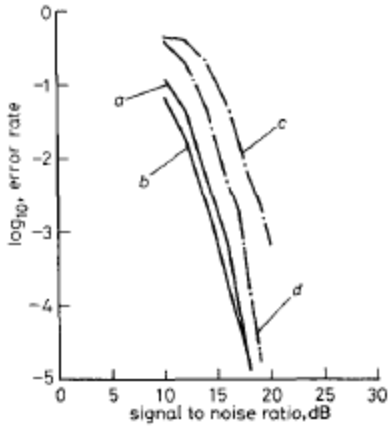


Fig.5.12 Simulation results showing relative BER performance for (4, I) DFE with (9.3, 1) MLP structure.

(a) η **= 0.1,** α **= 0.3,** β **= 0.05**

(b) η **= 0.07,** α **= 0.3,** β **= 0.05**

LMS (4, 1) DFE structure (c) μ **= 0.05 (d)** μ **= 0.035**

5.11 Discussion

The conventional structure of the DFE consists of a feed forward equalizer and a feedback filter, where the feed forward equalizer is linear. The linearity of the equalizer limits the performance of the system. Here a new approach for the DFE using multilayer structures is introduced. The back propagation learning algorithm is applied directly to the multilayer network. From comparison of simulation results it can be seen that the multilayer perceptron-based DFE provides better BER performance, especially in poor signal to noise ratio conditions, also that BER performance degrades less owing to decision errors and is also less sensitive to learning gain variation. We conclude that the multilayer perceptron-based DFE offers a superior performance (higher resolution) as a channel equalizer to that of the conventional DFE, because of its ability to form complex decision regions with nonlinear boundaries. It should be noted, however, that the structures invoked here are considerably more complex than the conventional DFE.

5.12 Summary

This chapter describes a new approach for a decision feedback equalizer using the multilayer perceptron structure for equalization in digital communications systems. Results indicate that the perceptron based decision feedback equalizer provides better bit error rate performance relative to the least mean square decision feedback equalizer, especially in high noise conditions, also that bit error rate performance degrades less owing to decision errors and is also less sensitive to gain variation.

Chapter 6.

Decision Feedback Equalization Using MLP and RBF

6. Decision Feedback Equalization Using MLP and RBF

6.1 Introduction

Channel equalization is an important subsystem in a communication receiver. Equalization is a technique used to remove inter-symbol interference (ISI) produced due to the limited bandwidth of the transmission channel [1]. When the channel is band limited, symbols transmitted through will be dispersed. This causes previous symbols to interfere with the next symbols, yielding the ISI. Also, multipath reception in wireless communications causes ISI at the receiver. Thus, equalizers are used to make the frequency response of the combined channel-equalizer system flat.

Two classes of equalizers are known: linear and non-linear equalizers. An example of the latter type is the decision feedback equalizer (DFE).The equalization process can be divided into two modes-a training mode and a decision-directed mode. In the first mode, the equalizer is trained to produce the expected output, by sending a training sequence and the coefficients of the equalizer are adjusted to produce the required output at each sampling time. In the second mode, the equalizer is operated on the channel to be equalized to estimate the channel output. The second mode is the normal operating mode in a practical communication system. Since equalization technique is simply deciding on a symbol from signals available in the signal space,(1 or -1 for the binary phase-shift keying(BPSK) system), it can be considered as a classification problem[6]. It accepts the delayed received samples as inputs, and outputs its decision which is one of the possible signals. In the M-ary case, the system has M different possible classes at the equalizer output.

In this thesis, an RBF net is used as a DFE. The architectures of the DFE and the RBF net is discussed. Then the use of RBF net to implement a DFE is presented. Simulation results are then discussed. Finally, conclusions and suggestions for future work are presented.

6.2 Decision Feedback Equalizers (DFEs)

A schematic diagram of a DFE is shown in Fig.6.1. An (n, m) DFE denotes an equalizer with n tapped delayed inputs and m feedback signals. So, m output samples are fed back to the input through a feedback filter in addition to the input samples. This feedback helps the system to de -correlate the noise that is produced by the ISI at the final output [10]. DFEs are usually implemented using LMS or RLS algorithms [1].In all cases, the input-output relation is expressed in the following equation (6.1)[1]:

$$I_k = \sum_{j=1}^{n} c_j \, x_{k-j} + \sum_{j=1}^{m} g_j \, y_{k-j} + c_e \, e_k \qquad (6.1)$$

Where, I_k is the output of the filter, x_k is the received signal, y_k is the decided symbol at the equalizer output. Also, c_j , g_j and c_e are the coefficients of the feed forward and feed backward filters. The error signal e_k , is the difference between the equalized signal y_k and the output of the equalizer I_k.The subscript $(k-j)$ in both filters indicates that the samples are shifted in the line at each sampling interval. Both the feed forward and the feed backward filters are considered as finite impulse response (FIR) filters. Equation (6.1) describes the function of the combined filters as an infinite impulse response (IIR) filters.

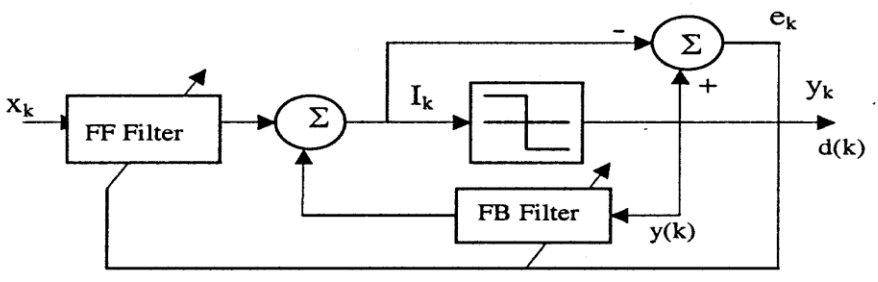


Fig.6.1 DFE using two FIR filters, one as feed forward and another as feed backward

Since the DFE is considered to be a non-linear equalizer, it is used more often than linear equalizers, especially for the case of severe –ISI channels. These channels are characterized in their frequency response by the existence of frequency nulls that make them totally non-linear and produce disturbed output [4].

The performance of DFEs depends on the number of the delayed inputs and the number of the feedback signals from output to input. It can be improved by feeding an error signal (the difference between the expected output and the produced output) back to the input in addition to the normal feedback signals [5].

6.3 Radial Basis Function (RBF)

RBF nets are well suited to solve interpolation problems. Such problems are stated as follows:

Given a set of input vectors { x_i} and the corresponding output vectors{ y_i}, find the appropriate transfer function that can fit noisy input vectors to produce the most appropriate output according to the given input/output vector pairs[3].It is clear that the equalization problem is a typical interpolation problem.

A general architecture of an RBF net is shown in Fig.6.2.It consists of two layers with the activation functions in the first layer being radial, and in the output layer being linear. The activation function of the first layer is called the basis function. It is a radial function characterized by being monotonically increasing or decreasing from a centre value [9].Examples of radial function are the thin plate spline, multi-quadratic, inverse multi-quadratic and the Gaussian functions [3]. The Gaussian function is most commonly used because of its smooth characteristics. It is given by (6.2) [7]:

$$h(x) = e^{-\frac{(x-c)^2}{r^2}} \qquad (6.2)$$

Where, c is the centre of the function and r is its spread constant. The centre and the spread constant control the location and the spread of the decision region of the radial function, respectively. The spread constants should be chosen such that the functions cover their areas and some of the adjacent areas in the space, increasing the ability of the ANNs to generalize for noisy patterns [13]. The output of the RBF net is given by (6.3):

$X = \sum_{j=1}^{p} x_j$ (6.3)

Where $y = \sum_{j=1}^{n} w_j \; h_j$(X)

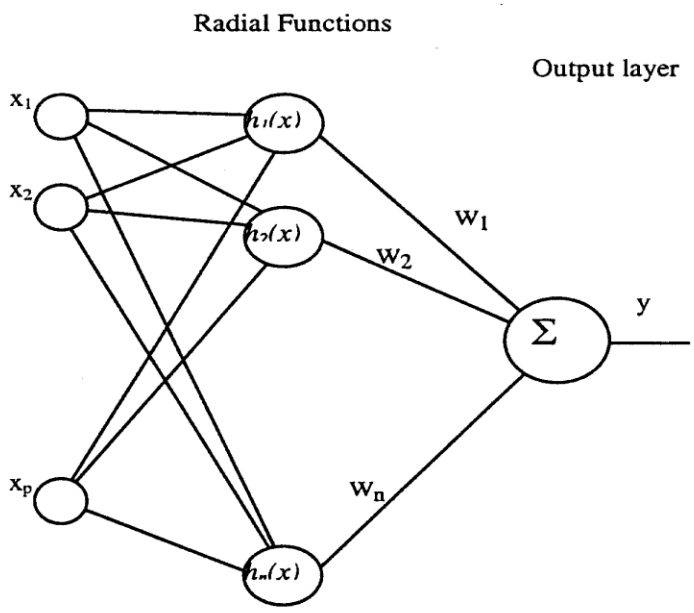


Fig.6.2 General architecture of an RBF net.

The basic idea behind the RBF development is Cover's theorem [7]. It says that complex pattern – classification problems are more likely to be linearly separable in high dimensional than in low- dimensional space. Using Gaussian radial functions in the RBF net converts problems into new ones in higher dimensional space. The RBF net is trained by presenting the training data vectors and the corresponding output vectors to

the net and it will compute its weight matrix that minimizes the cost function C given by [3]:

$$C = \sum_{i=1}^{p} \{ y_i - \sum_{j=1}^{n} w_j \, h_j \, (X_i) \}^2$$

These calculations are repeated by adding one basis function at a time until the required MSE is reached. When the RBF is trained using the exact interpolation method, the no. of basis functions needed is the same as the no. of examples used in training. This makes the ANN need more computations because of the large number of basis functions used [7, 13]. The training process used in this process is the one used in MATLAB. It uses the minimum no. of basis functions that are able to solve the problem undertaken with the required MSE [7]. Of course, for a given no. of training examples, the no. of basis functions used in this method is less than the no. of training examples [7, 13]. This improves the generalization abilities of the RBF not because using a no. of basis functions equal to the no. of examples makes the ANN unable to draw decisions for noisy examples; that may be encountered later during the operation mode [13].

6.4 THE IMPLEMENTED SYSTEM

The implemented RBF-based DFE consists of a tapped delay line that has 5 taps. At each sampling interval, the signals in the line are shifted by one location and a new received signal is put at the first tap. The RBF net is trained using 500 training samples with their corresponding outputs. It is initialized with one neuron whose activation function is Gaussian with a spread constant of 0.7. Each time, the RBF computes the weight matrix and adds one neuron if the MSE is still high. This process is repeated until the required MSE is obtained. The hidden layer consists of 170 and 300 basis functions for the DFE and linear equalizers, respectively. These nos. are the minimum nos. of basis functions needed to solve the equalization problem in each case and to have a MSE of

10^{-4}. The RBF based DFE is compared with a MLP based DFE that consists of a (9, 3, 1) MLP. This means that there are 9, 3 and 1 neurons in the input, hidden and output layers, respectively. The nine input signals constitute a delay line of 9 taps. Both the hidden and the output layers have activation functions of the tan-sigmoid shape. The MLP net is initialized using the first training example from the channel. The training process then continues using the back propagation algorithm with a variable training rate. Upon receiving a new training example, it computes the MSE and updates its coefficients accordingly. This process is repeated recursively until the required MSE, which was set to 10^{-4}, is achieved.

The two RBF and MLP –based DFEs are used to equalize two channels that are of practical importance. The first is a linear channel that introduces small distortions to its input [1]. The second is a severe-ISI channel whose frequency response has a deep null [4]. The latter type is faced often in practical communication systems and is very difficult to equalize using linear equalizers. However, they can be equalized efficiently using non linear equalizers such as DFEs. The two channels used are shown in Fig.6.3.

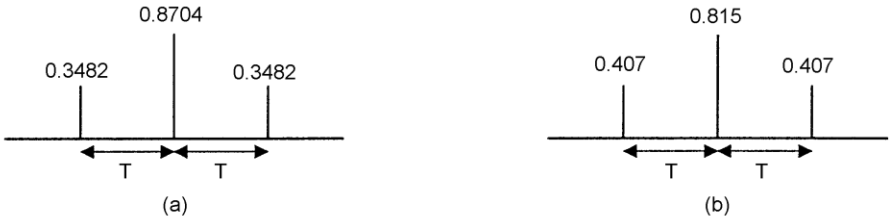


Fig.6.3 (a) Channel-1

(b)Channel-2

Two DFE cases were simulated. The first case is a DFE in which the detected symbols are used as feedback signals. In the second case, the correct symbols are used as feedback signals, which is not possible practically. This is because if the correct symbols are known to the receiver, there is no need for

doing communication [4]. However, it is used to find the lower bound of the performance of the DFE used. In summary, (5, 0) and (4, 1) DFEs are implemented using both MLP and RBF nets

6.5 SIMULATION RESULTS

The results of using linear equalization for channels 1 and 2 are shown in Figs. 6.5(a) and 6.5(b) respectively. The RBF based equalizer performance is better than that of the MLP based by 5 and 4 dBs, for channels 1 and 2, respectively at 10^{-2} bit error rate (BER). It is clear that channel 2 was not equalized well using linear equalization because of its severe ISI.

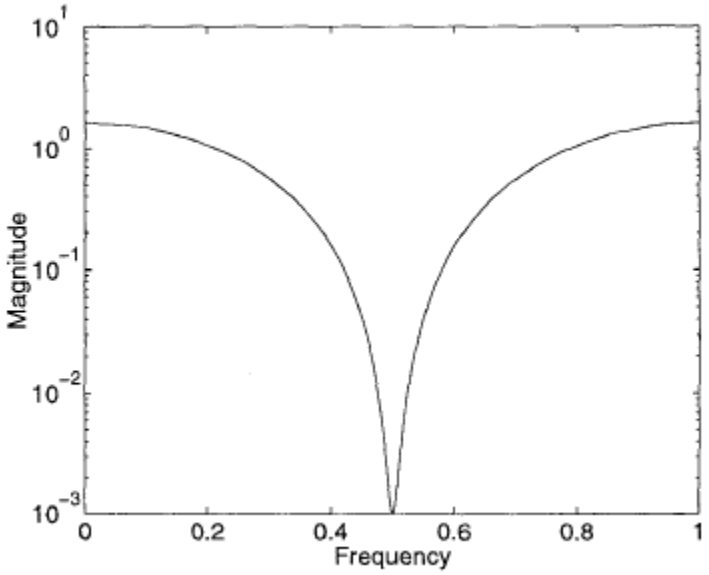
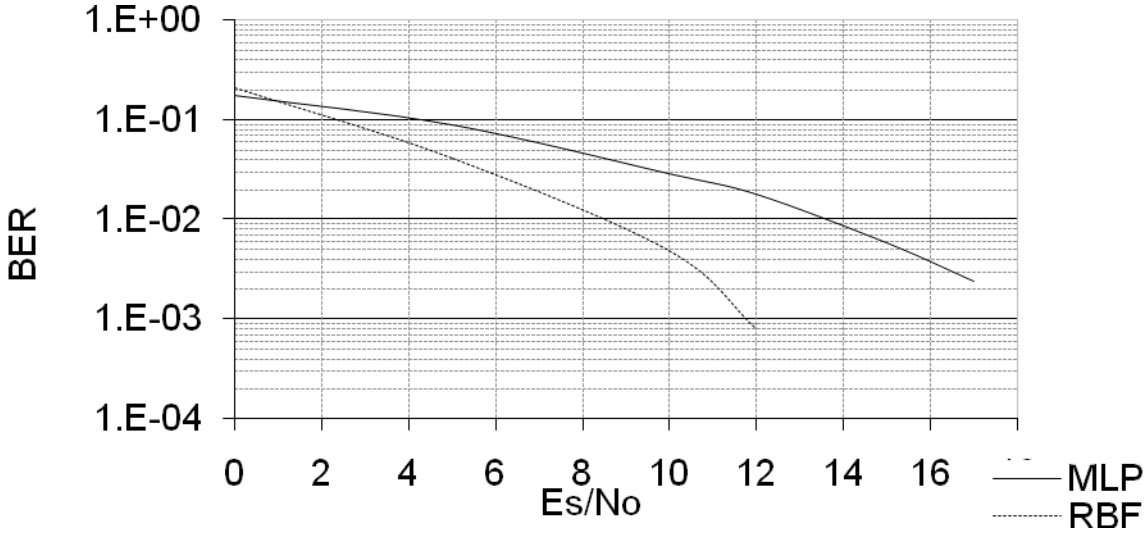
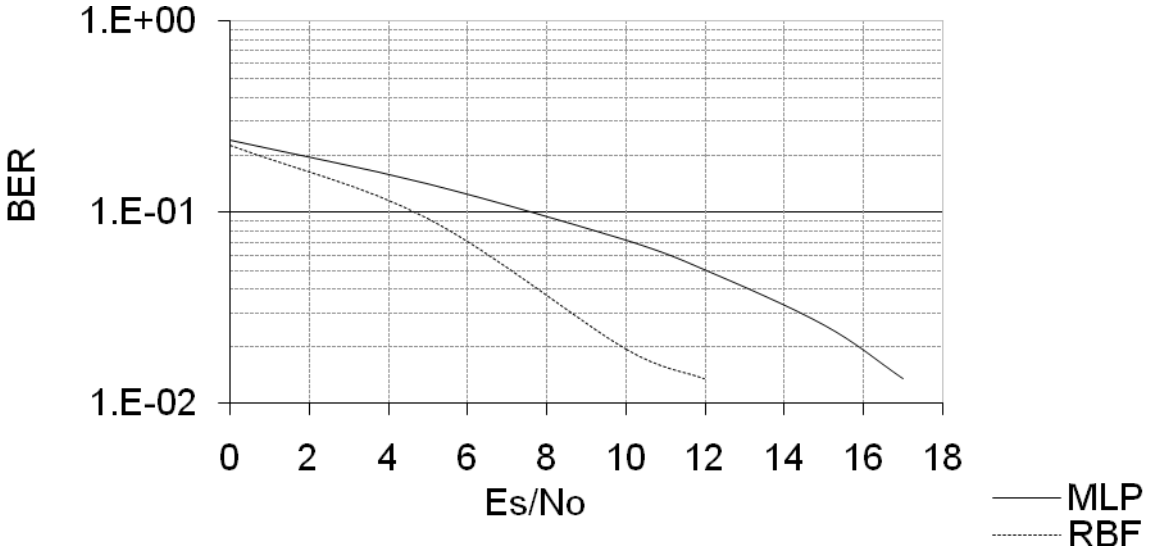


Fig.6.4 Frequency response of channel-2 showing deep null.



(a)

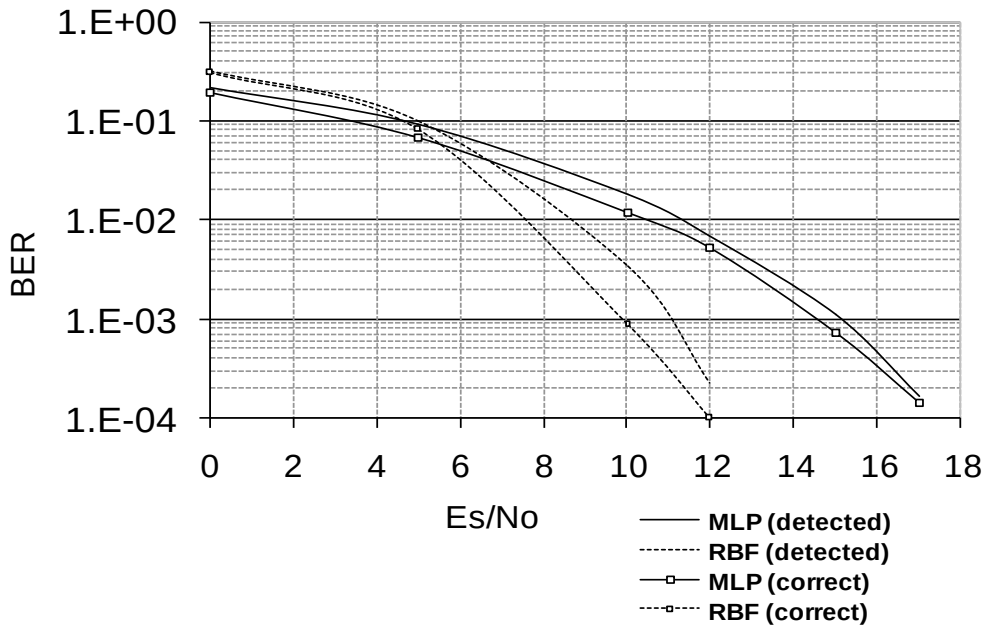
Fig. 6.5(a) Performance of linear equalization of Channel 1



(b)

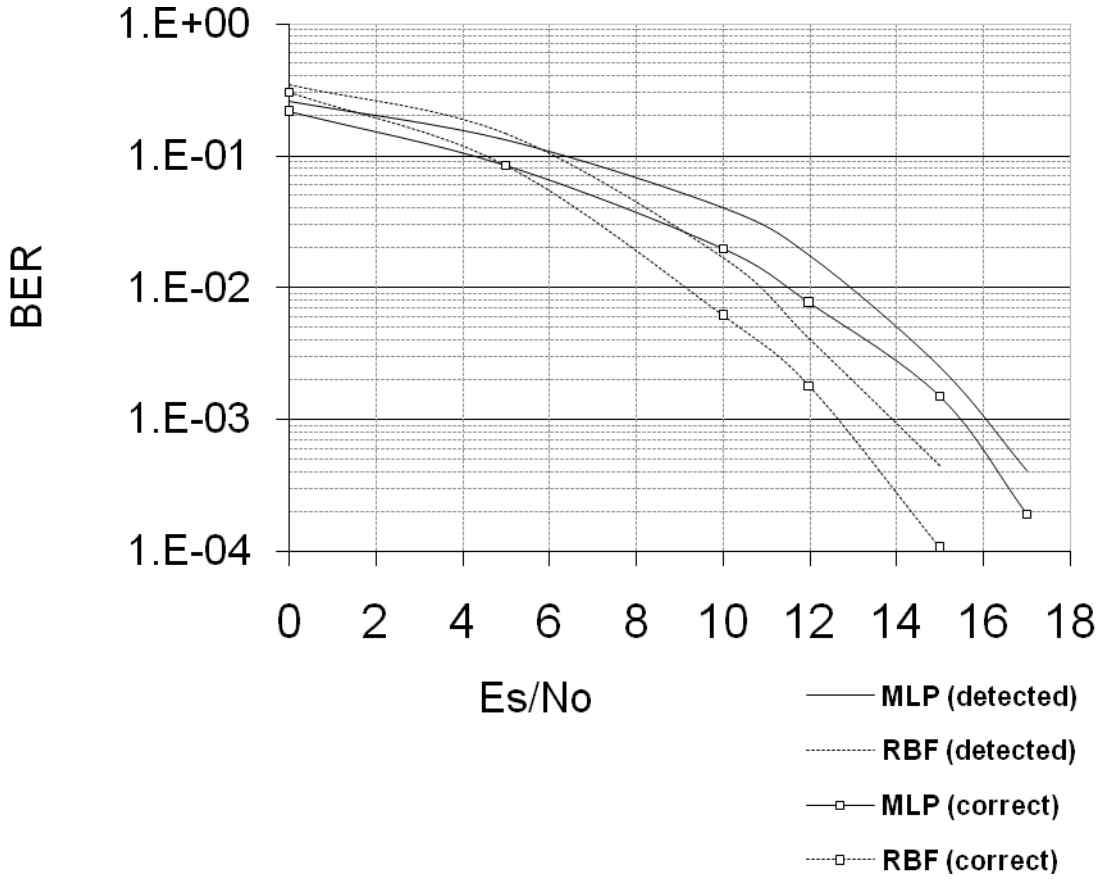
Fig.6.5 (b) Performance of linear equalization of Channel 2

Fig.6.6 (a) shows the performance of both MLP and RBF based (4, 1) DFEs for channel1. It is clear that the RBF based equalizer outperforms the MLP based one by about 4 dBs at 10^{-3} BER. Fig.6.6(b) shows the same information as part (a) for channel 2. Also, the RBF based DFE outperforms the DFE based on MLP by about 2dB. Of course, the overall performance for channel 2 is worse than that of channel 1 because channel 2 is more severe. In both channels, the DFE based on RBF is better than the one based on MLP even when the correct symbol is fed back in the MLP and the detected one is fed back in the RBF. This means the former DFE is better than the latter always, since feeding back the correct symbol is the most ideal case.



(a)

Fig.6.6 (a) Performance of DFE of channel 1



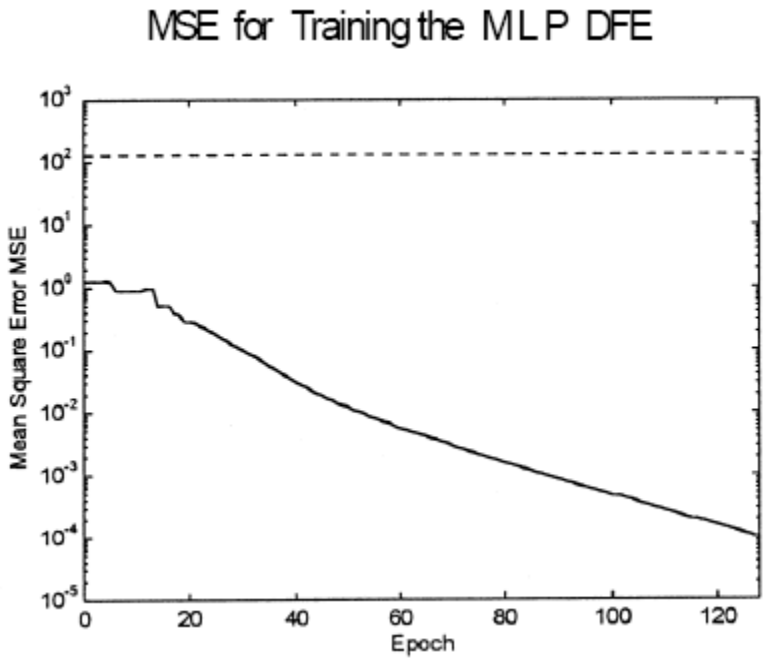
(b)

Fig. 6.6 (b) Performance of DFE of channel 2

Figs.6.7 (a) and 6.7(b) shows the convergence of both MLP and RBF-based DFEs, respectively. Both equalizers were able to reach the required MSE but the RBF is faster. On the other hand, the RBF based DFE needs more computations in the decision directed modes.

This is due to the high no. of basis functions in the hidden layer of the RBF system compared to the MLP system.

Simulation results showed that increasing the no. of neurons in the hidden layer of the MLP will not improve the convergence time or the BER performance. So, the price paid for reducing the BER and speeding up the training process by using the RBF based DFE, is the more computations required in the decision directed mode.



(a)

Fig. 6.7(a) Convergence of MLP-based DFE.

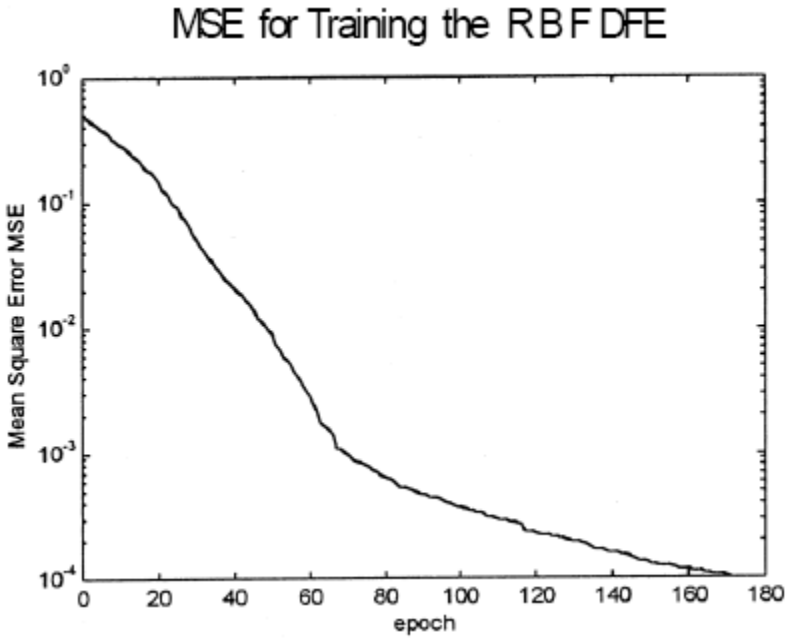


Fig.6.7 (b) Convergence of RBF-based DFE.

6.6 DISCUSSION AND CONCLUSION

In this thesis, linear and DFE equalizers were implemented using both MLP and RBF nets. The above systems were tested for two different channels. Results showed that linear equalizers are not good for severe-ISI channels. Also, it is seen that the RBF based equalizers perform better than the MLP based one, especially at high SNR. Moreover, the RBF equalizer converges faster than the MLP in the training mode but need more computational time in the decision directed mode, because of its large no. of neurons compared with the MLP. Trade off between fast convergence and performance on one side and the on line computational time on the other side should be taken into consideration upon designing such systems in practice.

The DFE performs better when the correct symbol is the feedback signal that is an ideal case. They also are efficient in reducing the effect of the deep frequency null of channel 2. According to [1], the MLP based DFE outperforms

the conventional DFE based on LMS and so does the RBF-DFE implemented in this thesis.

CHAPTER 7

Conclusion and Future Work

7. Conclusion and Future Work

7.1 Conclusion

The aim of this thesis is to find a proper artificial neural network (ANN) model for decision feedback equalization (DFE) and to compare it with conventional DFE using LMS algorithm. The prime advantages of using ANN models are their ability to learn based on optimization of an appropriate error function and their excellent performance for approximation of nonlinear functions.

Two ANN structures, namely, MLP and RBF nets along with LMS algorithm are discussed. Both MLP using BP algorithm and LMS algorithm are characterized by slow rate of convergence as these algorithms operates entirely on the basis of first-order information. RBF based DFE performs better than that of MLP based DFE and LMS based DFE as implemented in this thesis.RBF converges faster than that of MLP but there is a tradeoff between fast convergence in one side and the on line computational time in the other side should be taken into consideration upon designing such systems in practice.

7.2 Scope for Future Work

The scope of future work is outlined below:

1. Extension of this research is to implement the same concept using different training algorithms that converge faster.

2. Regarding the RBF net, regularization terms can be added to its weight matrix equation. It is claimed in [7] that this can reduce the noise variance in the output signal, which improves the performance.

3. Also, DFE can be implemented using error feedback as in [5], but via the ANN approach.

BIBLIOGRAPHY:

[1] QURESHI, S.U.H.: 'Adaptive equalization', Proc. IEEE, 1985, 73, (9), pp. 1349-1387.

[2] GIBSON, G.J., SIU, *S.,* and COWAN, C.F.N.: 'Multi-layer perceptron structures applied to adaptive equalizers for data communications'. IEEE Proceedings ICASSP Glasgow, Scotland, May 1989, pp. 1183-1186.

[3]Mulgrew B." Applying Radial Basis Function". IEEE signal Processing Mag. (March 1996)50-65.

[4] J. Proakis Digital communication NY.

[5] Kim , D. , Han , S., Eun, M., Choi , J and Choi , Y." An adaptive decision feedback equalizer using error feedback. " IEEE Trans. On Consumer Electronics 42, No.3 august 1996.

[6] Raffaele, P, Claudio, E Orlandi, G and Rao, B."Fast adaptive digital equalization by Recurrent Neural Networks." IEEE Trans. On Signal Processing, 45, No.11. (Nov.1997)

[7] Haykin S Neural Networks. A comprehensive foundation, 2nd Ed NY Prentice Hall Inc., 1998.

[8] Kirkland W and Taylor "On the application of Feedforward NN to channel equalization". Proc. IJCNN Int. Joint conf. N N New York 1992.

[9] Peng M, Niklas C and Proakis, "Adaptive equalization with N N: New multi layer Perception structure and their evaluation." Proc. ICASSP' IEEE Int. Conf. Decoust, Speech signal Processing, New York 1992.

[10] Kechriotis, G Zervas E and Manolakos E. " Using RNN for Adaptive communication channel equalization." IEEE Trans. Neural Networks, 5(March 1994).

[11] Chang, P, Yeh, B and Chang, C."Adaptive Packet equalization for indoor Radio channel using MLP." IEEE Trans, Vec.Technol.43 (August 1994).

[12] Al -Mashouq K and Reed, I." The use of neural nets to combine equalization with decoding for severe ISI channels". IEEE, Trans Neural Network (November 1994)

[13] Lee, J., Beach, C and Tepedelenliogh, N."Channel equalization using RBF Network." IEEE Int. conf. on Neural Networks 1996.

[14] P. S. R. Diniz Adaptive Filtering: Algorithms and Practical Implementation Kluwer Academic Publishers 1997.

[15] M. Austin, "Decision- feedback equalization for digital communication over dispersive channels, "M. I. T. RES. Lab. Electron., Tech. Rep. 461, August, 1967.

[16] B. Widrow and S. D. Stearns, Adaptive Signal Processing, Prentice-Hall Inc., 1985.

[16] D. A. George, R. R. Bowen, and J. R. Storey, " An adaptive decision feedback equalizer, " IEEE Trans. Commun., Vol. COM-19, no.3, pp. 281-293, June 1971.

[17] MINSKY, M., and PAPERT, S.: 'Perceptron: An introduction to computational geometry' (MIT Press. 1988)

[18] RUMELHART, DE., and McCLELLAND, J.L.: 'Parallel distributed processing: explorations in the microstructure of cognition' (MIT Press, 1986)

[19] LIPPMANN, R.P.: 'An introduction to computing with neural nets', IEEE ASSP Magazine, April 1987.4. (2)

[20] LAPEDES, A., and FARBER, R.: 'Nonlinear signal processing using neural networks: prediction and system modelling'. Los Alamos National Lab., Preprint La-Ur-87-2662

[21]D. A. George, R. R. Bowen, and J. R. Storey, " An adaptive decision feedback equalizer, " IEEE Trans. Commun., Vol. COM-19, no.3, pp. 281-293, June 1971.

[22] WIELAND, A., and LEIGHTON, R.: 'Geometric analysis of neural network capabilities'. 1st International Conference on Neural Networks. IEEE. June 1987, pp. 385-393

[23] COUCH 11, L.W.: Digital and analog communication systems' (Macmillan Publishing Co., Inc., 1983) Ch. 8, pp. 403-457.

[24] WIDROW, B., WINTER, R.G., and BAXTER, R.A.: 'Layered neural nets for pattern recognition', IEEE Trans. ASSP, July 1988, 36, pp. 1096-1099.

[25] R. W. Lucky. Automatic Equalization for Digital Communications. Bell Systems Technical Journal, 44:547–588, April 1965.

[26] R. Price. Nonlinearly feedback-equalized PAM vs. capacity for noisy filter channels.
In Rec. Int. Conf. Communications, Philadelphia, PA, 1972.

[27] P. Monsen. Feedback equalization for fading dispersive channels. IEEE Trans.Information Theory, 17:56–64, January 1971.

[28] J. Salz. Optimum mean-square decision feedback equalization. Bell System Technical Journal, 52(8):1341–1373, October 1973.

[29] D. Messerschmitt. A Geometric theory of intersymbol interference. Part I: Zero forcing and decision feedback equalization. Bell System Technical Journal, 52(10):1483–1519, November 1973.

[30] C. A. Belfiore and J. H. Park, Jr. Decision feedback equalization. Proc. IEEE, 67(8):1143–1156, August 1979.

[31] N. Al-Dhahir and J. M. Cioffi. MMSE Decision-Feedback Equalizers: Finite-Length Results. IEEE Trans. Information Theory, 41(4):961–975, July 1995.

[32] S.Siu,"Non-linear adaptive equalization based on multi-layer perceptron architecture," Ph.D. dissertation, University of Edinburgh, 1990.

[33] S. S.Chen, B.Mulgrew, "Adaptive bayesian equalizer with decision feed-back," IEEE Trans Signal Processing, vol. 41, no. 9, pp. 2918-2927, Sept 1993.

[34] O.Macchi, Adaptive processing, the least mean squares approach with applications in transmission. West Sussex. England: John Wiley and Sons,1995.

[35] X. I. Reed, Error control coding for data networks. Boston: Kluwer academic publishers, 1999.

[36] G.Forney, "Maximum-likelihood sequence estimation of digital sequences in the presence of inter-symbol interference," IEEE Trans, inform, theory,vol. 18, pp. 363-378, 1972.

[37] W. H.Fan, “A new adaptive iir filter," IEEE Trans. Circuits and Systems, vol. 33, no. 10, pp. 939-947, Oct 1986.

[38] K.C.Ho and Y.T.Chan, “Bias removal in equation-error adaptive iir filters," IEEE Trans. Signal Processing, vol. 43, no. 1, Jan 1995.

[39] J.E.Cousseau and P.S.R.Diniz, “New adaptive iir filtering algorithms based on the steiglitz-mcbride method," IEEE Trans. Signal Processing, vol. 45, no. 5, May 1997.

[40] I. M. W. K.L.Blackmore, R.C.Williamson, “Online learning via congregational gradient descent," Mathematics of Control, Signals, and Systems, vol. 10, pp. 331-363, 1997.

[41] J. M. F.Albu, A. Mateescu and B. Dorizzi, “Adaptive channel equalization using neural network," in Proc. IEEE on Telecommunications symposium, vol. 2, Aug 1998, pp. 438-441.